Generalizations and Applications of the Stochastic Block Model to Basketball Games and Variable Selection Problems

by

Lu Xin

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Statistics

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Name: Ji Zhu Title: Professor
Supervisor	Name: Mu Zhu Title: Professor
Supervisor	Name: Hugh Chipman Title: Professor
Internal Member	Name: Mary Thompson Title: Professor
Internal Member	Name: Pengfei Li Title: Associate Professor
Internal-external Member	Name: Pascal Poupart Title: Associate Professor

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Over the past decade, there has been an explosion of network data in a vast number of circumstances, such as the World Wide Web, social networks, gene interactions, economic networks, etc. Scientific analysis of networks is of significant importance in revealing laws governing complex systems. Community detection, one of the fundamental problems in network analysis, discovers the underlying cluster structure of nodes in a network. The Stochastic Block Model (SBM) is an influential framework for model-based community detection. In this thesis, we first propose a Continuous-time Stochastic Block Model (CSBM). Furthermore, we develop a multistate CSBM and use it to analyze Basketball games. Finally, we propose a novel variable selection approach by constructing networks among variables and applying SBM techniques.

Various Stochastic Block Models have been developed for static networks, such as a network of Facebook users. Theoretical properties of these models have been studied recently. However, for transactional networks, for example, a network of email users who frequently send emails to each other, research is relatively limited. Most existing works either do not take time into account or treat time in a discrete manner (as in discrete-time Markov chains). In contrast, we propose a Continuous-time Stochastic Block Model (CSBM) for transactional networks. Transactions between pairs of nodes are modeled as inhomogeneous Poisson processes, where the rate function of each Poisson process is characterized by the underlying cluster labels of the corresponding pair of nodes. The CSBM is capable of not only detecting communities but also capturing how transaction patterns evolve among communities.

As an important application, a multistate CSBM is developed and applied to basketball games. Basketball data analysis has gained enormous attention from enthusiasts and professionals from various fields. We advocate that basketball games can be analyzed as transactional networks. The multistate CSBM models basketball plays as continuous-time Markov chains. The model clusters players according to their playing styles and performance. It also provides cluster-specific estimates of the effectiveness of players at scoring, rebounding, stealing, etc, and also captures player interaction patterns within and between clusters. Moreover, the model reveals subtle differences in the offensive strategies of different teams. Applications to NBA basketball games illustrate the performance of the multistate CSBM.

The SBM framework can also be employed for variable selection. In the past two decades, variable selection has become one of the central topics in statistical learning and high-dimensional data analysis. Numerous methods have been successfully developed. In general, there are mainly three types of approaches: penalized likelihood methods, variable screening methods and Bayesian variable selection methods. Nevertheless, in this thesis, we propose a novel variable selection method: Variable Selection Networks, which is in a new framework — Variable Selection Ensembles. Given a regression model with p covariates, we consider the ensemble of all $p(p-1)/2$ submodels with two covariates. We construct networks with nodes being the p variables and each edge being a measure of the importance of the pair of variables to which it connects. We show that such networks have block structures. Variable selection is conducted by applying SBM techniques to these networks.

Acknowledgements

Foremost, my deep gratitude goes to my supervisors and mentors Dr. Mu Zhu and Dr. Hugh Chipman. Without their generous support, patient guidance and enormous trust, this thesis could not have been completed. They have helped me discover a brand new world as well as a brand new self, and led me through my path of becoming a Doctor of Philosophy, not only in Statistics but also in life. I will forever cherish the time with them and feel privileged to be their student.

I also would like to thank my thesis committee, Dr. Mary Thompson, Dr. Pengfei Li, Dr. Pascal Poupart and Dr. Ji Zhu for their detailed comments and insightful discussions. They have taught me the last class of my PhD study, from which I can benefit a lot in the future.

To my great office mates, Jiaxi Liang and Marco Shum, and my dear friends, Alex Bishop, Min Chen, Liqun Diao, Andrei Fajardo, Jenny Flagler George, Linval George, Feng He, Zhiyue Huang, Daniel Severn, Xichen She, Hua Shen, Chunlin Wang, Benjamin Waterman, Junko Takanashi Waterman, Chengguo Weng, Ying Yan, Yujie Zhong, thank you for the colorful, meaningful and memorable days.

Moreover, I would like to express my appreciation to Mary Lou Dufton for all her help during my entire graduate study in the department. I also want to thank Dr. Paul Marriott and Dr. Changbao Wu for their support as graduate chairs.

Last but not least, I am deeply grateful to my parents, Xiaobin Xin and Fangjun Li, for their limitless caring and unconditional love. Their kindness and integrity teach me the

great nature of human beings. Finally, I can not express enough thanks to my beautiful wife Celia Huang. Her beautiful heart, with the warmest encouragement and love, has helped me through the toughest days. Her beautiful mind, with infinite thoughts and passion, inspires me every single day. All in all, her beautiful smile lights up my life.

Dedication

To my parents Xiaobin Xin and Fangjun Li.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Abstract iv

Acknowledgements vi

Dedication viii

List of Tables xiii

List of Figures xvi

1 Introduction 1

1.1 Networks 1

1.2 Community Detection and Stochastic Block Models 5

1.3 Expectation-Maximization Algorithm 7

1.4 Spectral Clustering 11

1.5 Organization and Summary of Contributions 14

2 A Continuous-time Stochastic Block Model **17**

2.1 Introduction 17

2.1.1 Transactional Networks and Stochastic Block Models 18

2.1.2 Inhomogeneous Poisson Process 19

2.2 A Continuous-time Stochastic Block Model 21

2.2.1 Conditional Likelihood 22

2.2.2 An EM Algorithm 23

2.3 A Simulation Example 27

2.4 Summary and Remarks 28

3 Basketball Networks **32**

3.1 Introduction 32

3.1.1 An Overview of Basketball Analytics 33

3.1.2 Basketball Networks 35

3.2 A Multistate CSBM for Basketball Networks 38

3.2.1 Pseudo Conditional Likelihood 39

3.2.2 An EM$^+$ Algorithm 51

3.3 Applications to NBA data 57

3.3.1 Model simplifications and adjustments of the EM$^+$ algorithm 57

3.3.2 Miami Heat versus Boston Celtics in 2012 60

3.3.3 Cleveland Cavaliers versus Golden State Warriors in 2015 68

3.3.4 LeBron James: Miami Heat versus Cleveland Cavaliers 81

3.4 Summary and Remarks . . . 85

4 Variable Selection Networks **88**

4.1 Introduction . . . 88

4.2 A Binary Variable Selection Network . . . 90

4.2.1 Degree Distributions of the Binary VSN . . . 94

4.2.2 Correlations among Test Statistics . . . 99

4.3 A Weighted Variable Selection Network . . . 107

4.4 Variable Selection Network Algorithms . . . 108

4.5 An Iterative Group Screening Algorithm . . . 112

4.6 Simulation Study . . . 115

4.6.1 Simulation models . . . 116

4.6.2 Simulation results . . . 119

4.7 A Real Data Application . . . 129

4.8 Summary . . . 131

5 Summary and Future Research **133**

5.1 Summary of the Thesis . . . 133

5.2 Future Research . . . 136

5.2.1 Continuous-time Stochastic Block Models . . . 136

5.2.2 Basketball Networks . . . 137

5.2.3 Variable Selection Networks . . . 138

References **142**

APPENDICES **150**

A Some Details for the EM Algorithm in Section 3.2.2 **151**

A.1 The Conditional Expectation $\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z})|\mathbf{T}; \Theta^*]$ 151

A.2 Analytic Updates of Marginal Probabilities and Initial Probabilities 154

A.3 $\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z})|\mathbf{T}; \Theta^*]$ under Model Simplifications (3.21)-(3.22) 155

A.4 Analytic Updates of Transition Probabilities under Model Simplifications (3.21)-(3.22) 156

A.5 Confidence Bands for Estimated Rate Functions 158

B Proofs for Section 4.2 **160**

B.1 Proof of Lemma 1 160

B.2 Proof of Lemma 2 163

B.3 Proof of Theorem 1 170

B.4 Proof of Lemma 3 171

List of Tables

2.1 A transactional network 18

3.1 Two plays from game 1 of the 2012 NBA eastern conference finals between the Boston Celtics and the Miami Heat. The top three lines show one play for the Boston Celtics. The ball is inbounded to C#9 (Rajon Rondo) at time 0; Rondo dribbles the ball and passes it to C#5 (Kevin Garnett) at second 11; Garnett misses a 2-pointer shot at second 12. Lines 4 to 9 illustrate one play for the Miami Heat. 37

3.2 Clustering results for the 2011-2012 Miami Heat and Boston Celtics ($K = 3$). Cluster labels are C1, C2, C3. Three different clustering results are presented (two under "Alone" and one under "Together"). Player positions are included for reference only; they are not used by the clustering algorithm. 62

3.3 Estimated transition probabilities (P_{sk}) from each initial action to clusters C1, C2, C3, for three different clustering models of the 2011-2012 Miami Heat and Boston Celtics. 63

3.4 Estimated transition probabilities (P_{kl} and P_{ka}) for the 2011-2012 Miami Heat and Boston Celtics ($K = 3$). Rows are originating clusters and columns are receiving clusters and play outcomes. 66

3.5 Clustering results for the 2014-2015 Cleveland Cavaliers and Golden State Warriors ($K = 3$). Cluster labels are C1, C2, C3. Four different clustering results are presented (two teams $\times$ two games). Player positions are included for reference only; they are not used by the clustering algorithm. 70

3.6 Estimated transition probabilities (P_{sk}) from each initial action to clusters C1, C2 and C3, for four different clustering models of the 2014-2015 Cleveland Cavaliers and Golden State Warriors. 72

3.7 Estimated transition probabilities (P_{kl} and P_{ka}) for the 2014-2015 Cleveland Cavaliers and Golden State Warriors ($K = 3$). Rows are originating clusters and columns are receiving clusters and play outcomes. 78

3.8 Clustering results for the 2011-2012 Miami Heat and the 2014-2015 Cleveland Cavaliers together ($K = 4$). Cluster labels are C1, C2, C3, C4. Players appearing with two separate avatars for the clustering algorithm are bolded. Player positions are included for reference only; they are not used by the clustering algorithm. 82

3.9 Summary of differences between our work and others. 86

4.1 Simulation results for Model (a) 120

4.2 Simulation results for Model (b) 122

4.3 Simulation results for Model (c) 124

4.4 Simulation results for Model (d) 125
4.5 Simulation results for Model (e) 126
4.6 Simulation results for Model (f) 127
4.7 Simulation results for Model (g) 128
4.8 Variables selected for the communities and crime data by the VSN methods. 131

List of Figures

1.1 A static network 2

2.1 A Poisson process 20

2.2 Simulation example: fitted rate functions vs. true rate functions 28

3.1 A basketball play. The ball is inbounded to player r at time 0; r passes the ball to i at time t_1; i passes the ball to j at time t_2; . . .; player i receives the ball at time t_{m-1} and passes it to r at time t_m; the play ends when player r scores 2 points at time $T < 24$ seconds. 35

3.2 Weighted basketball network of 16 NBA games between 16 teams (Fewell et al., 2012). Circles represent the five positions (point guard, shooting guard, small forward, power forward, and center), and rectangles represent start or end points of a play. The width of the edge is proportional to the frequency of the corresponding ball transitions. The most frequent transition directions, which sum up to 60%, are colored red. 36

3.3 Segments of a play that are related to the $i \to j$ process. The $i \to j$ process consists of the solid point and the solid segments. 43

3.4 Fitted rate functions for the 2011-2012 Miami Heat and Boston Celtics, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rate with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively. 64

3.5 Fitted rate functions for the 2014-2015 Cavaliers, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rates with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively. 74

3.6 Fitted rate functions for the 2014-2015 Golden State Warriors, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rates with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively. 75

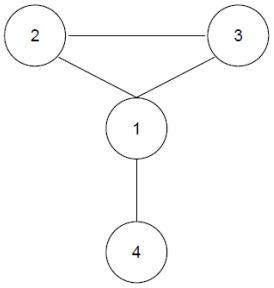
A.1 Rate functions displayed on top of each other in Figure 3.4 are displayed here individually with 95% pointwise confidence bands. 159

Chapter 1

Introduction

1.1 Networks

We live in a connected world. Overwhelmed by the meteoric rise of the internet and online social media, we can more vividly feel that the world is connected than we ever did. In fact, other than the internet and social networks, there are networks everywhere — neurons in the brain, genes, friendships, epidemics, economic networks, etc. In general, a network is simply a collection of objects connected to each other in a certain fashion. Mathematically, a network is a graph $G = (V, E)$, where V is a set of vertices/nodes and E is a set of edges. A network with n vertices can be represented by an $n \times n$ adjacency matrix, $\boldsymbol{A} = [A_{ij}]$, where $A_{ij} = 0$ or 1 indicates the absence or presence of the $i \to j$ edge, respectively. Figure 1.1 shows a simple undirected network ($A_{ij} = A_{ji}$ for all i and j) with four vertices and four edges. The relations between pairs of nodes do not have to be binary-valued. In



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 1.1: A static network

other words, entries of an adjacency matrix can be numbers other than 0 or 1. For real world networks, discovering their underlying structures and properties is of great interest. Indeed, researchers from numerous disciplines have paid tremendous attention to this over a long history.

As early as 1736, the great Leonhard Euler laid the foundation of graph theory by studying the *Seven Bridges of Königsberg*. Since then, graph theory has become an important branch of mathematics. In 1959, mathematicians Paul Erdös and Aflréd Rényi, and Edgar Gilbert independently introduced the first random graph model, the *Erdös-Rényi model* (Erdös and Rényi, 1959; Gilbert, 1959), which assumes that, given a fixed set of nodes, all pairs of nodes independently form edges with the same probability. In 1967, Stanley Milgram, a social psychologist, conducted an experiment that studied the degree of connectedness or chains of acquaintances of people in the United States, which led to the well-known *Six Degrees of Separation*. This phenomenon has lately been verified by many experiments. For example, a study on Microsoft Messenger instant messaging system shows that chains of contacts between users have an average length 6.6 (Leskovec and Horvitz, 2008). In 1998, Steven

Strogatz, an applied mathematician, and his student Duncan Watts show, in their paper on *Nature* (Watts and Strogatz, 1998), that many real world networks, including the power grid of the western U.S., the collaboration graph of film actors, and the neural network of the worm Caenorhabditis elegans have not only short average path lengths, but also high clustering coefficients, meaning that two connected neighbors of one node are likely to be connected or "friends of my friends are my friends". They introduced the *Watts-Strogatz Model* that captures both properties. In 1999, Albert-László Barabási, a physicist, and his student Réka Albert published their ground-breaking paper on *Science* (Barabási and Albert, 1999), showing the phenomenon that degree distributions of real world networks roughly follow a power law, i.e., the probability that a node has k connections is of the form $\mathbf{P}(k) \sim k^{-\gamma}$. They proposed the *Barabási–Albert Model* or *Preferential Attachment Model* to capture the power law degree distribution.

Meanwhile, statisticians and sociologists have been collaborating for more than a half century to develop statistical network models. Social network analysis has already become an important branch of social science. Long time ago, the phenomenon of high clustering coefficients, or "high transitivity" called by sociologists, had been investigated, for example, by Davis (1970). On the other hand, the preferential attachment is referred as the "Matthew effect" or "the rich get richer" in Sociology, which had also been studied, for example, by Price (1976). In the last two decades, numerous types of pervasive and fruitful statistical network models have been successfully developed and applied to study social behaviors and mechanisms. Two of the most influential frameworks are Latent Space Models and Exponential Random Graph Models. Latent space models assume existence of latent variables associated with vertices that determine the probabilities of edges among vertices.

Moreover, edges are usually assumed to be conditionally independent given the latent variables. There are different types of latent space models depending on the types of latent spaces and latent variables. For instance, Stochastic Block Models consider the scenario where the latent space is a discrete space and the latent variables are categorical variables. Distance models, such as the *Latent Space Social Network Model* (Hoff et al., 2002), the *Latent Ultrametric model* (Schweinberger and Snijders, 2003), assume that vertices can be projected into a latent metric space with a certain distance measure. They also assume that the probability of two vertices being connected depends on the distance of the two vertices in the latent metric space. Another very influential framework is the Exponential Random Graph Model (ERGM). It considers the following type of probability mass function of a network $\boldsymbol{A}$, $\mathbf{P}_\theta(\boldsymbol{A}) = \exp(\sum_i \theta_i s_i(\boldsymbol{A}))/\kappa(\theta)$, where $s_i(\boldsymbol{A})$ can be a statistic of the network or a covariate, and $\kappa(\theta)$ is the normalization term. The *Markov Exponential Random Graph model*, a special ERGM that has been comprehensively studied, assumes that two edges are conditionally independent given the other edges. Such an assumption implies that the configurations of statistics in the probability mass function, i.e., $S_i(\boldsymbol{A})$, must be the number of overall edges, the number of triangles or the number of k-star for any possible k. An overview of the ERGM can be found in Lusher et al. (2012). Dynamic networks have also been studied. Network dynamics mostly refer to state changes of edges or nodes, transactions among nodes, etc. The time scale for dynamics can be discrete or continuous. Many statistical models treat dynamic networks as Markov chains or Hidden Markov chains. Snijders (2011) and Kolaczyk (2009) provide helpful reviews of statistical network models.

The focus of this thesis is on the Stochastic Block Model. This will be reviewed in the

next section.

1.2 Community Detection and Stochastic Block Models

In real networks, nodes sometimes fall into different groups or communities, where nodes in the same community show a similar pattern in terms of forming edges, whereas nodes from different communities show distinct patterns. For example, in social networks, actors often have various backgrounds, e.g., students, businessmen, professionals, etc, and people with different backgrounds often have different social structures; in gene networks, genes from different units have different biological functions and thus interact with each other in different ways. Given a network, finding the underlying node communities can help us better understand the network structure and reveal network mechanisms. Indeed, community detection, one of the fundamental problems in network analysis, tries to search for natural clusters of network nodes. It differs from traditional graph partitioning algorithms in that the number and size of clusters are not pre-specified, but depend on the network. In fact, determining the number of communities is still an open problem in community detection (Saldana et al., 2016; Chen and Lei, 2016; Wang and Bickel, 2016).

There are different types of community detection approaches. Some methods try to conduct community detection purely by algorithms, for example, the *Hierarchical Clustering method* defines similarity measures for nodes and assembles nodes with high similarity together. Some methods define certain global criteria and search for the community realization that optimizes them, for example, the *modularity* (Newman, 2003). Finally,

model-based methods impose probabilistic assumptions on networks and adopt statistical inference to conduct community detection. The Stochastic Block Model (SBM) is one of the most influential model-based community detection frameworks. In the past decade, many generalizations of the SBM have been developed and their theoretical properties have been investigated.

Given a single network or static network $\boldsymbol{A}$, the Stochastic Block Model (Snijders and Nowicki, 1997) assumes that nodes belong to different blocks/communities, and nodes in the same block are stochastically equivalent. The distribution of an edge between two nodes is governed only by the blocks to which the nodes belong. Given the block labels of all nodes, all edges are independent with each other.

More explicitly, consider a network with n nodes and assume they are from K blocks. Given the node labels $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$, where $e_i \in \{1, 2, \ldots, K\}$, the edges are assumed to be independent Bernoulli random variables with

$$\mathbf{P}(A_{ij} = 1 | e_i, e_j) = P_{e_i e_j}, \tag{1.1}$$

where $\{P_{kl} : k, l = 1, 2, \ldots, K\}$ are K^2 parameters. The conditional distribution of the entire network, given the node labels, is of the form

$$\mathbf{P}(\boldsymbol{A}|\mathbf{e}) = \prod_{0 \le i,j \le n} P_{e_i e_j}^{A_{ij}} (1 - P_{e_i e_j})^{1 - A_{ij}}. \tag{1.2}$$

Given a network, our goal is to find the "best" label configuration $\mathbf{e}$. It can be obtained by maximizing the *profile likelihood function* (Bickel and Chen, 2009), which is the function

(1.2) with the estimated probabilities $\{\hat{P}_{kl} : k, l = 1, 2, \ldots, K\}$ plugged in. Finding the optimal solution is NP-hard. However, heuristic algorithms are available, for example, a label switching algorithm (Zhao et al., 2012). We may also use an Expectation-Maximization algorithm (Snijders and Nowicki, 1997) or spectral clustering algorithms (Jin, 2015; Lei and Rinaldo, 2015). Note that, all methods above work only when K, the number of clusters, is given. In practice, K needs to be determined. As mentioned previously, many research works have been devoted to this problem very recently (Saldana et al., 2016; Chen and Lei, 2016; Wang and Bickel, 2016).

Many generalizations have been developed for the standard SBM. The *Degree-corrected Stochastic Block Model* (Karrer and Newman, 2011) relaxes the strong assumption that nodes in the same block are exactly stochastically equivalent by adding individual strength parameters for nodes. Another influential work is the *Mixed Membership Stochastic Block Model* (Airoldi et al., 2008), which allows one node to belong to different blocks when communicating with different nodes. Theoretical properties of SBMs, such as detection consistency, have been investigated by Bickel and Chen (2009); Zhao et al. (2012); Jin (2015); Lei and Rinaldo (2015) and others.

1.3 Expectation-Maximization Algorithm

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is adopted in Chapter 2 and Chapter 3 for model fitting, so we briefly introduce it in this section.

The EM algorithm is widely used for dealing with missing data or latent variables. Suppose

$\mathbf{X}$ is observed data and $\mathbf{Z}$ is missing data or latent variables. Let $\boldsymbol{l}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta})$ denote the complete log-likelihood function of the model and $\boldsymbol{\theta}$ denote the parameters. The EM algorithm iterates over the following two steps:

E-Step Take conditional expectation of $\boldsymbol{l}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta})$ given $\mathbf{X}$ under $\hat{\boldsymbol{\theta}}$, the estimated parameters from the last step. That is

$$Q(\boldsymbol{\theta}) = \mathbf{E}[\boldsymbol{l}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta})|\mathbf{X}; \hat{\boldsymbol{\theta}}]. \tag{1.3}$$

M-Step Update parameters by calculating

$$\hat{\boldsymbol{\theta}}^{(new)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}). \tag{1.4}$$

Example: A classic application of the EM algorithm is to estimate Gaussian Mixture models. A mixture of K Gaussians is

$$f(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2), \tag{1.5}$$

where $\{\mathcal{N}(x; \mu_k, \sigma_k^2) : k = 1, 2, \ldots, K\}$ are the probability density functions of the *Gaussian components* and $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$ are the *mixing coefficients* with constraints $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. We can think that a sample from such a mixture of Gaussian is generated in two steps: first, draw a component label k from $\{1, 2, \ldots, K\}$ with probability $\{\pi_1, \pi_2, \ldots, \pi_K\}$, respectively; second, draw a sample from $\mathcal{N}(x; \mu_k, \sigma_k^2)$.

Given N samples from the Gaussian mixture distribution (1.5), denoted by $\mathbf{X}$, the likeli-

hood function is

$$\mathcal{L}(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2) \}. \tag{1.6}$$

The log-likelihood is

$$\ln \mathcal{L}(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2) \}. \tag{1.7}$$

Clearly, due to the summation inside the logarithm, it is hard to directly maximize the likelihood over the parameters. The problem can be solved by the EM algorithm.

We introduce latent variables $\boldsymbol{z}_n = (z_{n1}, z_{n2}, ..., z_{nK}), n = 1, 2, ..., N$ to indicate the component labels of the N samples, respectively, such that

$$z_{nk} = \begin{cases} 1, & \text{if sample } n \text{ belongs to component } k; \\ 0, & \text{otherwise.} \end{cases} \tag{1.8}$$

Assume $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_N$ to be i.i.d. $multinomial(1, \boldsymbol{\pi})$ random variables, with $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_K)$. The complete distribution function of $\mathbf{X}$ and $\mathbf{Z}$ is

$$\mathcal{L}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{\pi}) = \mathcal{L}(\mathbf{X}|\mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}) \cdot \mathcal{L}(\mathbf{Z}; \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2) \right]^{z_{nk}}. \tag{1.9}$$

Note that $z_{nk} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{nk} = 1$, which implies that, for a realization of $\boldsymbol{z}_n$, only one of the entries is 1 and all the others are 0. Therefore, the marginal distribution function

of $\mathbf{X}$ is given by

$$\sum_{\mathbf{Z}} \mathcal{L}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{\pi}) = \sum_{\mathbf{Z}} \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2)\right]^{z_{nk}} = \prod_{n=1}^{N} \Big\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2) \Big\}, \tag{1.10}$$

which is the same as (1.6).

The conditional distribution of $\mathbf{Z}$ given $\mathbf{X}$ is

$$\mathcal{L}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}) = \prod_{n=1}^{N} \frac{\prod_{k=1}^{K} \left[\pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2)\right]^{z_{nk}}}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2)}. \tag{1.11}$$

We can see that given $\mathbf{X}$, indicators $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N$ are conditionally independent and

$$\mathbf{E}(z_{nk}|\mathbf{X}) = \mathbf{P}(z_{nk} = 1|\mathbf{X}) = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n; \mu_j, \sigma_j^2)}. \tag{1.12}$$

Hence, the EM algorithm is as follows:

E-step Take log of the complete likelihood (1.9)

$$\boldsymbol{l}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \Big\{ z_{nk} \big[\ln \pi_k + \ln \mathcal{N}(x_n; \mu_k, \sigma_k^2) \big] \Big\}. \tag{1.13}$$

Take conditional expectation of the log-likelihood given the observed data, under the parameters estimated in the last step,

$$\begin{aligned} &\mathbf{E}_{\mathbf{Z}}[\boldsymbol{l}(\mathbf{X}, \mathbf{Z}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \boldsymbol{\pi})|\mathbf{X}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma^2}}, \hat{\boldsymbol{\pi}}] \\ &\qquad = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{E}_{\mathbf{Z}} \big\{ z_{nk} \big[\ln \pi_k + \ln \mathcal{N}(x_n; \mu_k, \sigma_k^2) \big] | \mathbf{X}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma^2}}, \hat{\boldsymbol{\pi}} \big\} \end{aligned}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\mathbf{E}(z_{nk}|\mathbf{X};\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\sigma}^{\mathbf{2}}},\hat{\boldsymbol{\pi}})\left[\ln\pi_k + \ln\mathcal{N}(x_n;\mu_k,\sigma_k^2)\right]\right\}$$
$$= \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\frac{\hat{\pi_k}\mathcal{N}(x_n;\hat{\mu_k},\hat{\sigma_k}^2)}{\sum_{j=1}^{K}\hat{\pi_j}\mathcal{N}(x_n;\hat{\mu_j},\hat{\sigma_j}^2)}\left[\ln\pi_k + \ln\mathcal{N}(x_n;\mu_k,\sigma_k^2)\right]\right\}. \quad (1.14)$$

M-step Maximize (1.14) over all parameters. Let $\gamma(z_{nk}) = \mathbf{P}(z_{nk} = 1|\mathbf{X};\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\sigma}^{\mathbf{2}}},\hat{\boldsymbol{\pi}})$ and $N_k = \sum_{n=1}^{N}\gamma(z_{nk})$. We have the following solutions:

$$\hat{\mu_k}^{(new)} = \frac{\sum_{n=1}^{N}\gamma(z_{nk})x_n}{N_k}, \quad (1.15)$$
$$\hat{\sigma_k}^{2(new)} = \frac{\sum_{n=1}^{N}\gamma(z_{nk})(x_n-\hat{\mu_k})^2}{N_k}, \quad (1.16)$$
$$\hat{\pi_k}^{(new)} = \frac{N_k}{N}. \quad (1.17)$$

Thanks to the closed-form solutions, we can simply update the parameters by iteratively operating the **M-step**.

1.4 Spectral Clustering

Spectral clustering methods are popular for clustering nodes in networks with block structures (Rohe et al., 2011; Jin, 2015; Lei and Rinaldo, 2015). In Chapter 4, we adopt the spectral clustering algorithm introduced by Lei and Rinaldo (2015) to conduct variable selection upon variable selection networks. Therefore, we introduce the algorithm in this section, with derivations closely following Lei and Rinaldo (2015).

Consider an undirected network with symmetric adjacency matrix $\boldsymbol{A} = [A_{ij}]$, where A_{ij} is a random variable for $i \neq j$ and $A_{ii} = 0$ for all $i = 1, 2, \ldots, p$. Suppose the nodes are from K underlying clusters. Moreover, given any two nodes i and i' from cluster k and any two nodes j and j' from another cluster l, we have

$$\mathbf{E}(A_{ij}) = \mathbf{E}(A_{i'j'}) := B_{kl}, \tag{1.18}$$

where $\{B_{kl} : k, l = 1, 2, \ldots, K\}$ are parameters. Essentially, the expectation of an edge is determined by clusters of the nodes to which it connects, so all edges actually fall into $K(K-1)/2$ blocks and the edges in the (k, l) block have the same expectation B_{kl}.

Let $\boldsymbol{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iK})^T$ indicate the cluster label of node i, where

$$z_{ik} = \begin{cases} 1, & \text{if node } i \text{ belongs to cluster } k, \\ 0, & \text{otherwise.} \end{cases}$$

Suppose there are p nodes. Define $\boldsymbol{Z}_{p\times K} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_p)^T$ and $\boldsymbol{B}_{K\times K} = [B_{kl}]$. According to the previous description, we have

$$\mathbf{E}(\boldsymbol{A}) = \boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^T - diag(\boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^T). \tag{1.19}$$

Define $\Delta = diag(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_K})$, where p_k is the number of nodes in cluster k, and we obtain

$$\boldsymbol{Q} := \boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^T = \boldsymbol{Z}\Delta^{-1}\Delta\boldsymbol{B}\Delta\Delta^{-1}\boldsymbol{Z}^T. \tag{1.20}$$

Suppose the eigen-decomposition of $\Delta \boldsymbol{B} \Delta$ is $\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T$, then

$$\boldsymbol{Q} = \boldsymbol{Z}\Delta^{-1}\Delta\boldsymbol{B}\Delta\Delta^{-1}\boldsymbol{Z}^T = \boldsymbol{Z}\Delta^{-1}\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T\Delta^{-1}\boldsymbol{Z}^T, \tag{1.21}$$

which can be verified to be the eigen-decomposition of $\boldsymbol{Q}$, and K eigen-vectors are columns of $\boldsymbol{Z}\Delta^{-1}\boldsymbol{E}$.

Let $\boldsymbol{U} = \boldsymbol{Z}\Delta^{-1}\boldsymbol{E}$. It is easy to see that $\boldsymbol{U}$ only has K distinct rows. In fact, suppose node i is in cluster k, the i^{th} row of $\boldsymbol{U}$ is $\boldsymbol{z}_i^T\Delta^{-1}\boldsymbol{E}$, which returns the k^{th} row of $\Delta^{-1}\boldsymbol{E}$.

According to (1.19), $\mathbf{E}(\boldsymbol{A})$ and $\boldsymbol{Q}$ are the same except diagonal elements, so the top K eigenvectors of $\mathbf{E}(\boldsymbol{A})$ should approximately equal to the K eigenvectors of $\boldsymbol{Q}$, which have K distinct rows. Furthermore, it is reasonable to expect that the top K eigenvectors of the observed network $\boldsymbol{A}$ share a similar pattern as that of $\mathbf{E}(\boldsymbol{A})$. Therefore, we obtain the following spectral clustering algorithm for networks with block structures.

1. Calculate $\hat{\boldsymbol{U}} \in \mathbb{R}^{p\times K}$, which consists of the leading K eigenvectors of $\boldsymbol{A}$ (ordered in absolute eigenvalue) .

2. Conduct k-means clustering on the rows of $\hat{\boldsymbol{U}}$, with K clusters.

From the derivation (1.19) to (1.21), we can see that the above algorithm can be applied to both binary networks and weighted networks with arbitrary valued edges, as long as $\mathbf{E}(\boldsymbol{A})$ has a block structure.

1.5 Organization and Summary of Contributions

This thesis considers generalizations and applications of the Stochastic Block Model to basketball games and variable selection problems.

In Chapter 2, a Continuous-time Stochastic Block Model is proposed for transactional networks. We first introduce the *transactional network*, a special type of dynamic network that records transactions among nodes over a period of time. Our goal is community detection for the nodes in a transactional network. A natural idea is to generalize the standard Stochastic Block Model. Indeed, some research works have been devoted to this recently. However, most existing works either do not take time into account or treat time in a discrete manner (as in discrete-time Markov chains). In contrast, we propose a Continuous-time Stochastic Block Model (CSBM). Transaction processes are modeled as inhomogeneous Poisson processes, where the rate function of a transaction process between a pair of nodes depends only on the underlying communities of the nodes. Cubic B-splines are used to model the rate functions. We develop an EM algorithm to fit the CSBM. Finally, we illustrate the model by a simulation example.

In Chapter 3, a multistate CSBM is developed and applied to analyze basketball games. We first provide an overview of basketball data analysis, a field growing rapidly in the past few years. In short, traditional analysis mainly focuses on the box score, which lists statistics of players and teams of each game, for example, number of field goals made, number of rebounds, etc. Recently, thanks to much richer real-time data, researchers have gone beyond the box score. Some researchers consider basketball games as networks

and others model basketball plays as stochastic processes. We combine these two ideas and provide a novel perspective for basketball analysis. In particular, we advocate that basketball games can be analyzed as transactional networks in the sense that players are nodes and ball passes are transactions. Our interest is to cluster players into different groups according to their playing styles. A multistate CSBM is developed for basketball networks, where each basketball play is modeled as a Markov chain. The transition rate functions of the Markov chain depend on the latent cluster labels of players. To fit the multistate CSBM, we develop an EM$^+$ algorithm, which is an EM algorithm followed by a complementary heuristic algorithm. At the end of this chapter, the model is illustrated by appealing applications to NBA games. Our paper based on this research project (Xin et al., 2016) has been accepted by *the Annals of Applied Statistics*.

In Chapter 4, a novel variable selection method, Variable Selection Networks (VSN), is proposed. For variable selection, researchers have mainly focused on three types of approaches in the last two decades: penalized likelihood methods, variable screening methods and Bayesian variable selection methods. The VSN does not belong to any of these three categories. Instead, it advocates a different framework, Variable Selection Ensembles (VSE). The main idea of the VSE is to evaluate an ensemble of submodels and use the aggregate information to select variables. Given p covariates, the VSN considers the ensemble of all $p(p-1)/2$ submodels with two covariates. By treating each variable as a node and the importance measure of each pair of variables as an edge between them, such an ensemble of submodels is actually a network. In this chapter, we first construct variable selection networks for the $p < n$ case and investigate their theoretical properties. We show that such networks have block structures. Algorithms incorporating Stochastic

Block Model techniques are developed to conduct variable selection for variable selection networks. Moreover, for the $p \geq n$ case, we propose an iterative group screening method to reduce the number of variables. Finally, the VSN is compared to many state-of-the-art and newly developed variable selection methods by simulations. The VSN is very competitive in comparison to existing approaches.

In Chapter 5, we summarize the thesis and discuss future research.

Chapter 2

A Continuous-time Stochastic Block Model

2.1 Introduction

This chapter presents a basic Continuous-time Stochastic Block Model and an EM algorithm to fit the model. This is the first step of our generalization of the standard Stochastic Block Model (Section 1.2), so we only illustrate the model and the algorithm by a simple simulation example. As such, this chapter is not a standalone project and, unlike Chapters 3 and 4, there is no plan to publish this chapter on its own. Following the foundations built in this chapter, we will develop a complex multistate Continuous-time Stochastic Block Model for basketball networks in Chapter 3, where we explicitly illustrate the model using real basketball data.

2.1.1 Transactional Networks and Stochastic Block Models

Under certain circumstances, instead of simply observing a connection between each pair of nodes, we observe a series of transactions, for example, phone calls among a number of people in a period of time. Such networks are called transactional networks. The corresponding data, as shown in Table 2.1, simply records senders, recipients and time of transactions.

Table 2.1: A transactional network

From	To	Time of transaction
1	4	03/29/2015, 08:27
1	7	03/29/2015, 09:01
3	1	03/30/2015, 17:11
⋮	⋮	⋮

In general, the transactional network is one special kind of longitudinal networks. Longitudinal networks, which record the evolution of networks over a series of time points, have been studied by researchers for more than two decades. A typical way to model longitudinal networks is by discrete or continuous-time Markov chains, for example, the *actor-oriented models* (Snijders, 1996). The rate functions of Markov chains are usually modeled as functions of network statistics. Snijders (2001) provides a brief overview of this class of models. Recently, inspired by event history analysis (Cook and Lawless, 2007), Vu et al. (2011) model transactions among nodes as recurrent events, where the intensity functions are modeled as multiplicative and additive functions of cumulative network statistics.

To conduct community detection for transactional networks, many recent works adopt the SBM framework. Shafiei and Chipman (2010) focus on the number of transactions, but do not consider the time factor. Ho et al. (2011), and Xu and Hero (2014) study networks at discrete time points and use State Space Models to describe intertemporal dynamics.

In this Chapter, we propose a Continuous-time Stochastic Block Model (CSBM), which models transactions over time as inhomogeneous Poisson processes. In the next chapter, we propose a multistate CSBM for basketball networks, where ball transactions are modeled as Continuous-time Markov chains. For both CSBM models, the rate functions are governed by the underlying communities of nodes, and they are fitted by cubic B-splines. EM algorithms are developed to estimate the CSBMs.

DuBois et al. (2013) have similar ideas to ours, but they focus on generic transactional networks and parameterize rate/intensity functions using network statistics. Their model can not be directly applied to the multistate case such as basketball networks. Another difference is that they use MCMC to fit their model.

2.1.2 Inhomogeneous Poisson Process

We will model transactions in transactional networks as inhomogenous Poisson processes. Hence, we first look at the distribution of an inhomogeneous Poisson process, often used in event history analysis (Cook and Lawless, 2007). Figure 2.1 shows a Poisson process with m events, happening at times $t_1 < \cdots < t_m$ over the interval $[t_0, t_m]$. Suppose that our observation of the process stops at time t_m. Let $\rho(t)$ denote the rate function of this

Figure 2.1: A Poisson process

inhomogeneous Poisson process. The distribution is of the form

$$\mathcal{L} = \prod_{i=1}^{m} \rho(t_i) \cdot \exp\Big(- \int_{t_0}^{t_m} \rho(u)\mathrm{d}u\Big) \tag{2.1}$$

$$= \prod_{i=1}^{m} \Big(\rho(t_i) \cdot \exp\big(- \int_{t_{i-1}}^{t_i} \rho(u)\mathrm{d}u\big)\Big) \tag{2.2}$$

$$= \prod_{i=1}^{m} \mathcal{L}\big((t_{i-1}, t_i]\big). \tag{2.3}$$

The time intervals $\{(t_{i-1}, t_i],\ i = 1, 2, \ldots, m\}$ are independent. For each time interval $(t_{i-1}, t_i]$, the distribution consists of two parts: the part for the actual event, $\rho(t_i)$, and the part for the the time gap between events, $\exp\big(- \int_{t_{i-1}}^{t_i} \rho(u)du\big)$. The derivation of (2.1) is as follows, which closely follows the presentation by Cook and Lawless (2007, p. 30).

Let N_t denote the number of events in the time interval $[t, t + \Delta t)$. By the definition of the Poisson process, for a very small Δt,

$$\mathcal{P}(N_t = 0) = 1 - \rho(t)\Delta t + o(\Delta t), \tag{2.4}$$

$$\mathcal{P}(N_t = 1) = \rho(t)\Delta t + o(\Delta t), \tag{2.5}$$

$$\mathcal{P}(N_t \geq 2) = o(\Delta t). \tag{2.6}$$

Consider a partition of $[T_0, T)$, say $T_0 = u_0 < u_1 < u_2 \cdots < u_R = T$. By the "independent

increment" property of the Poisson process, we have

$$\mathcal{P}([T_0, T_1)) = \prod_{r=0}^{R-1} \mathcal{P}([u_r, u_{r+1})) = \prod_{r=0}^{R-1} \mathcal{P}(N_{u_r})$$
$$= \Big(\prod_{N_{u_r}=0} [1 - \rho(u_r)\Delta u_r + o(\Delta u_r)] \Big) \cdot \Big(\prod_{N_{u_r}=1} [\rho(u_r)\Delta u_r + o(\Delta u_r)] \Big) \cdot \Big(\prod_{N_{u_r}\geq 2} [o(\Delta u_r)] \Big). \quad (2.7)$$

Notice that $\log[1-\rho(t)\Delta t] = -\rho(t)\Delta t + o(\Delta t)$, so the logarithm of the first product in (2.7) — the one over $N_{u_r} = 0$ — approaches the Riemann integral, $-\int_{T_0}^{T} \rho(t)dt$, in the limit. Thus, dividing Δu_r into each respective term that corresponds to the interval $[u_r, u_{r+1})$ and taking the limit as $R \to \infty$ and consequently as $\Delta u_r = u_{r+1} - u_r \to 0$, we obtain that the desired distribution is

$$\prod_{i=1}^{m} \rho(t_i) \cdot \exp\Big[-\int_{T_0}^{T} \rho(u)du\Big].$$

Hence, we see that the probability distribution function of the Poisson process consists of two parts: the first part, $\prod_{i=1}^{m} \rho(t_i)$, corresponds to all event times; and the second part, $\exp\big[-\int_{T_0}^{T} \rho(u)du\big]$, corresponds to all time gaps.

2.2 A Continuous-time Stochastic Block Model

In this section, we develop a Continuous-time Stochastic Block Model for transactional networks. Recall the two principles of the Stochastic Block Model: the nodes in the same block are stochastically equivalent and all edges are conditionally independent given the

community labels of the nodes. In transactional networks, we observe transaction processes between pairs of nodes. Following the principles of the SBM, a Continuous-time SBM is constructed as follows:

- Assume that nodes belong to K different clusters, and each node only belongs to one cluster.
- Define K^2 rate functions $\{\rho_{kl}(t) : k, l = 1, 2, \ldots, K\}$. The transactions from node i to j are modeled as an inhomogeneous Poisson process with the rate function being $\rho_{e_i e_j}(t)$, where e_i and e_j are the cluster labels of i and j, respectively.
- Given the cluster labels of all nodes, transactions are conditionally independent.

2.2.1 Conditional Likelihood

Suppose there are n nodes with cluster labels $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ and the network is observed over a time period (T_0, T). Let t_{ijh} denote the h^{th} time of the transaction from i to j; $\boldsymbol{t}_{ij} = \{t_{ijh} : h = 1, 2, \ldots, m_{ij}\}$ denote the transaction time points from i to j, where m_{ij} is the total number of transactions from i to j; and $\mathbf{T} = \{\boldsymbol{t}_{ij} : i, j = 1, 2, \ldots, n\}$ denote all transaction times. According to the design, the conditional distribution of such a transactional network, given the cluster labels of nodes, is

$$\mathcal{L}(\mathbf{T}|\mathbf{e}) = \prod_{1 \le i,j \le n} \mathcal{L}(\boldsymbol{t}_{ij}|e_i, e_j) \tag{2.8}$$

$$:= \prod_{1 \le i,j \le n} \Big\{ \prod_{h=1}^{m_{ij}} \rho_{e_i e_j}(t_{ijh}) \cdot \exp\big[-\int_{T_0}^{T} \rho_{e_i e_j}(u)du\big] \Big\}, \tag{2.9}$$

where we utilize the distribution function of a Poisson process (2.1). For the sake of simplicity, we define

$$\mathcal{L}_{ije_ie_j} = \mathcal{L}(\boldsymbol{t}_{ij}|e_i, e_j), \tag{2.10}$$

which is the conditional distribution function of the transaction process from i to j given their cluster labels. In particular, if we know that cluster labels of i and j are $e_i = k$ and $e_j = l$, respectively, then

$$\mathcal{L}_{ijkl} = \mathcal{L}(\boldsymbol{t}_{ij}|e_i = k, e_j = l) = \prod_{h=1}^{m_{ij}} \rho_{kl}(t_{ijh}) \cdot \exp\big[-\int_{T_0}^{T} \rho_{kl}(u)du\big]. \tag{2.11}$$

We model the rate functions by cubic B-splines such that

$$\rho_{kl}(t) = \sum_{p=1}^{P} e^{\beta_{klp}} B_p(t), \tag{2.12}$$

where $\{B_1(t), B_2(t), \ldots, B_P(t)\}$ are basis functions and $\boldsymbol{\beta} = \{\beta_{klp} : k, l = 1, 2, \ldots, K; p = 1, 2, \ldots, P\}$ denote coefficients. The parametrization $e^{\beta_{klp}}$ ensures $\rho_{kl}(t) \geq 0$.

2.2.2 An EM Algorithm

The cluster labels $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ are latent variables, which are unknown. In this section, we develop an EM algorithm for the CSBM. The procedure is very similar to the EM algorithm for Gaussian mixture models (Section 1.3).

Let $\boldsymbol{z_i} = (z_{i1}, z_{i2}, \ldots, z_{iK})$ denote a latent label indicator for node i such that

$$z_{ik} = \begin{cases} 1, & \text{if node } i \text{ belongs to cluster } k; \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

Assume $\{\boldsymbol{z_i} : i = 1, 2, \ldots, n\}$ are marginally i.i.d. $multinomial(1, \boldsymbol{\pi})$, with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$. Let $\mathbf{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)$ and $\Theta = \{\boldsymbol{\beta}, \boldsymbol{\pi}\}$. The complete likelihood of the network is

$$\begin{aligned} \mathcal{L}(\mathbf{T}, \mathbf{Z}; \Theta) &= \mathcal{L}(\mathbf{T}|\mathbf{Z}; \boldsymbol{\Theta}) \cdot \mathcal{L}(\mathbf{Z}; \Theta) \\ &= \prod_{1 \le i \ne j \le n} \prod_{1 \le k,l \le K} (\mathcal{L}_{ijkl})^{z_{ik} z_{jl}} \cdot \mathcal{L}(\mathbf{Z}; \Theta) \\ &= \prod_{1 \le i \ne j \le n} \prod_{1 \le k,l \le K} \Big\{ \prod_{h=1}^{m_{ij}} \rho_{kl}(t_{ijh}) \cdot \exp\big[- \int_{T_0}^{T} \rho_{kl}(u) du \big] \Big\}^{z_{ik} z_{jl}} \cdot \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}}. \end{aligned} \tag{2.14}$$

Note that we assume one node does not communicate with itself. The log-likelihood is

$$\begin{aligned} & l(\mathbf{T}, \mathbf{Z}; \Theta) \\ &= \sum_{1 \le i \ne j \le n} \sum_{1 \le k,l \le K} \Big\{ z_{ik} z_{jl} \big[\sum_{h=1}^{m_{ij}} \log \rho_{kl}(t_{ijh}) - \int_{T_0}^{T} \rho_{kl}(u) du \big] \Big\} + \sum_{i=1}^{n} \sum_{k=1}^{K} (z_{ik} \log \pi_k). \end{aligned} \tag{2.15}$$

E-Step Take conditional expectations of latent variables in the log-likelihood given observed time points,

$$\begin{aligned} & \mathbf{E}[l(\mathbf{T}, \mathbf{Z}; \Theta) | \mathbf{T}; \Theta^*] \\ &= \sum_{1 \le i \ne j \le n} \sum_{1 \le k,l \le K} \Big\{ \mathbf{E}(z_{ik} z_{jl} | \mathbf{T}; \Theta^*) \cdot \big[\sum_{h=1}^{m_{ij}} \log \rho_{kl}(t_{ijh}) - \int_{T_0}^{T} \rho_{kl}(u) du \big] \Big\} \end{aligned}$$

$$+\sum_{i=1}^{n}\sum_{k=1}^{K}\big[\mathbf{E}(z_{ik}|\mathbf{T};\Theta^*)\log\pi_k\big], \tag{2.16}$$

where "$*$" indicates the parameters estimated from the last step.

The situation now is more difficult than Gaussian mixture models, because the latent variables $\{\boldsymbol{z_i} : i = 1, 2, \ldots, n\}$ here are not conditionally independent due to interactions of nodes. Therefore, it is infeasible to calculate the exact conditional expectations $\mathbf{E}(z_{ik}z_{jl}|\mathbf{T};\Theta^*)$ and $\mathbf{E}(z_{ik}|\mathbf{T};\Theta^*)$; for example, in order to calculate $\mathbf{E}(z_{ik}|\mathbf{T};\Theta^*)$, we need to marginalize the cluster labels of all nodes that interact with node i. Instead, we adopt *Gibbs Sampling* to sample from $\mathcal{L}(\mathbf{Z}|\mathbf{T};\Theta^*)$ and then use the corresponding sample means to approximate $\mathbf{E}(z_{ik}z_{jl}|\mathbf{T};\Theta^*)$ and $\mathbf{E}(z_{ik}|\mathbf{T};\Theta^*)$.

Gibbs Sampler Let $\mathbf{Z}^{-i} = \{\boldsymbol{z_j} : j \neq i\}$ denote the latent cluster indicators of all players other than i. The idea of the Gibbs sampler is to draw

$$\begin{aligned}
\mathbf{z}_1 &\sim \mathcal{L}(\mathbf{z}_1|\mathbf{Z}^{-1},\mathbf{T};\Theta^*),\\
\mathbf{z}_2 &\sim \mathcal{L}(\mathbf{z}_2|\mathbf{Z}^{-2},\mathbf{T};\Theta^*),\\
&\vdots\\
\mathbf{z}_n &\sim \mathcal{L}(\mathbf{z}_n|\mathbf{Z}^{-n},\mathbf{T};\Theta^*),\\
\mathbf{z}_1 &\sim \mathcal{L}(\mathbf{z}_1|\mathbf{Z}^{-1},\mathbf{T};\Theta^*),\\
\mathbf{z}_2 &\sim \mathcal{L}(\mathbf{z}_2|\mathbf{Z}^{-2},\mathbf{T};\Theta^*),\\
&\vdots
\end{aligned}$$

repeatedly until the stationary distribution is reached. Under the current parameter

estimate Θ^*, the conditional distribution of $\boldsymbol{z_i}$ given $\mathbf{Z}^{-i}$ and $\mathbf{T}$ is

$$\mathcal{L}(\boldsymbol{z_i}|\mathbf{Z}^{-i}, \mathbf{T}; \Theta^*) = \frac{\mathcal{L}(\mathbf{T}, \mathbf{Z}; \Theta^*)}{\sum_{\boldsymbol{z_i}} \mathcal{L}(\mathbf{T}, \mathbf{Z}; \Theta^*)}, \tag{2.17}$$

a multinomial distribution which is easy to sample from. More explicitly, suppose that, at the current step, $z_{jc_j} = 1$ for $j \neq i$ — this means $e_j = c_j$ for all $j \neq i$ or that c_j is the current group label for node j. Then, the conditional probability of node i belonging to cluster k is

$$\begin{aligned}
&\mathcal{P}(z_{ik} = 1|\mathbf{Z}^{-i}, \mathbf{T}; \Theta^*) \\
&\qquad = \mathcal{P}(e_i = k|\{e_j = c_j : j \neq i\}, \mathbf{T}; \Theta^*) \\
&\qquad = \frac{\mathcal{L}(\mathbf{T}, \mathbf{e} = (c_1, c_2, \ldots, c_{i-1}, k, c_{i+1}, \ldots, c_n); \Theta^*)}{\sum_{l=1}^{K} \mathcal{L}(\mathbf{T}, \mathbf{e} = (c_1, c_2, \ldots, c_{i-1}, l, c_{i+1}, \ldots, c_n); \Theta^*)}.
\end{aligned} \tag{2.18}$$

M-Step Maximize (2.16) with respect to parameters $\Theta = \{\boldsymbol{\beta}, \boldsymbol{\pi}\}$. It is easy to get the update equation for $\boldsymbol{\pi}$,

$$\pi_k^{(new)} = \frac{\sum_{i=1}^{n} \mathbf{E}(z_{ik}|\mathbf{T}; \Theta^*)}{\sum_{i=1}^{n} \sum_{l=1}^{K} \mathbf{E}(z_{il}|\mathbf{T}; \Theta^*)} = \frac{\sum_{i=1}^{n} \mathbf{E}(z_{ik}|\mathbf{T}; \Theta^*)}{n}, \tag{2.19}$$

where the second equation utilizes the fact that the conditional expectations are conditional probabilities. However, there is no closed-form solution for $\boldsymbol{\beta}$. We use the quasi-Newton method with L-BFGS-B updates — more specifically, we use the `optim` function in R and supply with it the analytic form of the gradient. In general, the EM algorithm can be trapped in a local solution. We leave the discussion on this issue in the next chapter when

we apply the EM algorithm to a more complex CSBM.

2.3 A Simulation Example

In this section, the CSBM and the EM algorithm are illustrated by a simulation example. We generated a transactional network with 20 nodes, which are from two communities with 10 nodes in each community. Rate functions are piecewise cosine functions:

within group 1

$$\rho_{11}(t) = (0.4\cos(\frac{2t\pi}{100}-\pi)+0.6)I(0 \leq t \leq 100)+(0.2\cos(\frac{2t\pi}{100}-\pi)+0.4)I(100 \leq t \leq 200), \tag{2.20}$$

within group 2

$$\rho_{22}(t) = (0.2\cos(\frac{2t\pi}{100}-\pi)+0.4)I(0 \leq t \leq 100)+(0.4\cos(\frac{2t\pi}{100}-\pi)+0.6)I(100 \leq t \leq 200), \tag{2.21}$$

and between group 1 and group 2

$$\rho_{12}(t) = \rho_{21}(t) = 0.1\cos(\frac{2t\pi}{100} - \pi) + 0.1. \tag{2.22}$$

For each pair of nodes, we generated a process of transactions according to the corresponding rate function over the time 0-200. Overall, the generated data has $20 \times 19/2 = 190$ processes and 11,605 transactions.

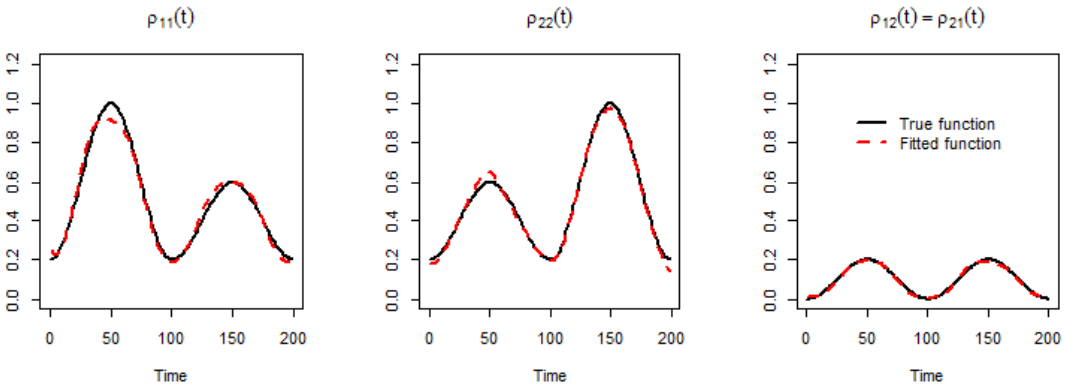


Figure 2.2: Simulation example: fitted rate functions vs. true rate functions

The EM algorithm converges quickly in about 5 iterations and can accurately cluster the nodes to two groups. We fix the starting points for all coefficients to be 0, while we start the Gibbs sampling by uniformly generating one label configuration from all configurations. The EM algorithm, when runs for multiple times, starts with pure random label configurations for the Gibbs sampling. For this simple example, we find that running the EM algorithm for multiple times yields the same result. Figure 2.2 shows the fitted rate functions vs. the true rate functions. The functions are fitted very well except small discrepancies at the first peaks of both $\rho_{11}(t)$ (left panel) and $\rho_{22}(t)$ (middle panel), probably due to our choice of using 11 basis functions for B-splines.

2.4 Summary and Remarks

In this chapter, we have proposed a Continuous-time Stochastic Block Model, a natural generalization of the standard SBM for transactional networks. Transactions between each

pair of nodes are modeled as an inhomogeneous Poisson process, with the rate function depending only on the community labels of the two nodes. We adopt B-splines to fit the rate functions. An EM algorithm is developed to estimate the model. The CSBM is illustrated by a simple simulation example.

Here, we make a few important remarks about the EM algorithm.

The underlying communities of nodes are indicated by $\mathbf{E}(\mathbf{Z}|\mathbf{T};\Theta)$, the conditional expectations/probabilities of the latent variables given transactions. We find that, in the E-step, the conditional probabilities driving the Gibbs sampler turn out to be fairly close to 0 or 1, that is, in equation (2.18), one of the K terms being summed in the denominator is significantly larger than the others. The reason is that each node is involved in many transactions. As far as the likelihood function is concerned, these transactions act as if they were repeated measurements, which reinforce the assignment of the node to a particular group.

To help understand the above issue, we take the Gaussian mixture scenario (Section 1.3) as an illustration. Recall that the conditional probability that point i belongs to cluster k is given by equation (1.12), copied below,

$$P(i \in \text{cluster } k|\mathbf{X};\boldsymbol{\mu},\boldsymbol{\sigma}) = \frac{\pi_k \mathcal{N}(x_i;\mu_k,\sigma_k^2)}{\sum_{l=1}^{K}\pi_l \mathcal{N}(x_i;\mu_l,\sigma_l^2)},$$

where $\mathbf{X}$ denotes all data points; $\boldsymbol{\mu} = \{\mu_k : k = 1, 2, \ldots, K\}$ and $\boldsymbol{\sigma} = \{\sigma_k : k = 1, 2, \ldots, K\}$ are means and standard deviations of the K Gaussians, respectively. Now if we have repeated observations for the random variable i, say $\{x_{ij} : j = 1, 2, \ldots, m_i\}$. The conditional

probability becomes

$$P(i \in \text{cluster } k|\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{\pi_k \prod_{j=1}^{m_i} \mathcal{N}(x_{ij}; \mu_k, \sigma_k^2)}{\sum_{l=1}^{K} \pi_l \prod_{j=1}^{m_i} \mathcal{N}(x_{ij}; \mu_l, \sigma_l^2)}, \tag{2.23}$$

which is closer to 0 or 1. If the sample size m_i is large, the conditional probability (2.23) would be considerably close to 0 or 1.

For the CSBM, equation (2.18) is similar to (2.23) in terms of structures. In addition, we expect the number of transactions to be very large in a transactional network, so each node is involved in many transactions. Hence, the Gibbs sampler (2.17) approximately follows a multinomial distribution with one entry of the probability vector being 1 and the others being 0. As a result, the Gibbs sampler converges very quickly to a singular probability mass. This essentially reduces the EM algorithm to something analogous to a k-means algorithm: the E-step re-assigns the nodes to different groups and the M-step re-estimates the parameters. Overall, the EM algorithm converges in just a few iterations.

In the end, we very briefly discuss about the computational complexity of the EM algorithm. Suppose the number of nodes is n, the number of clusters is k, the number of B-spline basis functions is p and the length of observation time is t.

First, the computational complexity for calculating the distribution function of an inhomogeneous Poisson process (2.1) is dominated by the numerical integration, which is roughly at least $O(tp)$. The number of events does not matter too much, because the numerical integration should evaluate much more points (depending on the accuracy level) than the event points over the time interval .

Second, for the E-step, the Gibbs sampling certainly costs most time. The computational complexity for getting one sample from $\mathcal{L}(\mathbf{Z}|\mathbf{T};\Theta^*)$ is approximately $n\times k\times(n-1)\times O(tp) = O(n^2ktp)$. Roughly speaking, $n \times k$ means going through all clusters for all nodes; then given one node with one cluster label, we need to evaluate the distribution functions of its transactions with all the other $n-1$ nodes, and thus the complexity is $(n-1) \times O(tp)$. As discussed, we do not need too many Gibbs samples in each iteration.

Finally, for the M-step, the most complexity clearly comes from the quasi-Newton method for updating the coefficients. Based on our experiences, the M-step costs much more time than the E-step. It is easy to see that the complexity of evaluating the log-likelihood function (2.16) is $O(n^2k^2tp)$. The complexity of the quasi-Newton method is certainly much more than that. It is hard to figure out the exact complexity of the quasi-Newton method, because it depends on the structure and shape of the log-likelihood function.

Chapter 3

Basketball Networks

3.1 Introduction

In this chapter, we present an important application of the Continuous-time Stochastic Model (CSBM). In particular, we construct a multistate CSBM and use it to analyze basketball games. We first give a brief overview of basketball analytics in Section 3.1.1. Then we provide a new perspective suggesting that basketball games can be analyzed as transactional networks in Section 3.1.2. A multistate CSBM together with an EM$^+$ algorithm are developed in Section 3.2. In Section 3.3, applications to NBA games illustrate that the multistate CSBM can reveal interesting insights of basketball games. We make some remarks in the end of the chapter.

3.1.1 An Overview of Basketball Analytics

For decades, basketball data analysis has gained enormous attention from basketball professionals and basketball enthusiasts from various fields. The top goal has always been to better understand how players and teams play, and conduct evaluations more efficiently and objectively. Over the last few years, the explosion of available data, the growth of computer power and the developments of statistical models have made complex modeling of basketball data possible. A revolution is happening in the field of basketball data analysis.

The traditional approaches focus on the box score, which lists the statistics of players and teams of each game, for example, number of field goals attempted, field goals made, rebounds, blocks, steals, plus-minus($+/-$), and other snapshot statistics. By combining the box score statistics, empirically or through regression analysis, various metrics have been developed to evaluate player and team performances (Oliver, 2004; Shea and Baker, 2013). However, "there is no Holy Grail of player statistics" (Oliver, 2004). As pointed out by Shea and Baker (2013), the metrics are either "bottom up" or "top down". Bottom-up metrics mostly focus on the individual performance, whereas top-down metrics put emphasis at the team level. Traditional box score metrics mostly fail to take into account two important factors of basketball: the interaction of players and the fact that a basketball play is a real-time process.

Recently, researchers have started to investigate basketball games from these two perspectives. By treating player positions (point guard, shooting guard, small forward, power forward and center) as network nodes and ball passes as network edges, Fewell et al. (2012)

advocate "Basketball is not a game, but a network". They illustrate ball transition patterns of different teams by their basketball networks. Additionally, they quantitatively analyze basketball games and teams by calculating network properties such as degree centrality, clustering coefficient, network entropy and flow centrality. However, when building the networks, Fewell et al. (2012) only consider the cumulative passes of games. Hence, the networks are not able to capture details of basketball plays. Neither can they describe players' individual performances. In 2013, the National Basketball Association(NBA) installed optical tracking systems (SportVU technology) in all thirty courts to collect real-time data. The tracking system records the spatial position of the ball and the positions of all players on the court at any time of the game. It also records all actions of the games. Using such comprehensive data, Cervone et al. (2016) model the evolution of a basketball play as a complex stochastic process. Their model reveals both offensive and defensive strategies of players and teams. Ultimately, the model estimates the expected scores an offensive team can make at any time of the play. The two approaches above certainly provide more insights and more accurate evaluations of players, teams and basketball plays.

In the NBA, teams obtain new players through trades, free agency and the annual draft. There are so many potential players, especially college players, that no scout is able to keep close track of all of them. Clustering players to a number of groups, according to their performance and playing styles, can efficiently narrow down the target space. When searching for players, basketball managers, scouts and coaches always hope that the new player can quickly fit in the current team. Therefore, how players interact with teammates is of great importance. This must be taken into account during the clustering procedure.

We propose a multistate Continuous-time Stochastic Block Model (CSBM) to address the problem of player clustering. We model basketball games as transactional networks and a basketball play as an inhomogeneous continuous-time Markov chain. The CSBM clusters the players according to their performances on the court. It also effectively reveals the players' play styles and the teams' offensive strategies.

3.1.2 Basketball Networks

We now look at basketball, a team game. Players pass the ball to each other and form networks, with players as vertices and passing as transactions on edges. A basketball game is made of basketball plays. Generally, a basketball play starts with inbounding, rebounding, or stealing the ball. During a play, the team with the ball plays offense and the other team plays defense. A play ends when the offensive team shoots the ball (scores or misses but the ball hits the rim), makes a turnover, or the offensive player is fouled when shooting the ball, etc. In the NBA, the time limit for one play is 24 seconds. Figure 3.1 illustrates one basketball play.

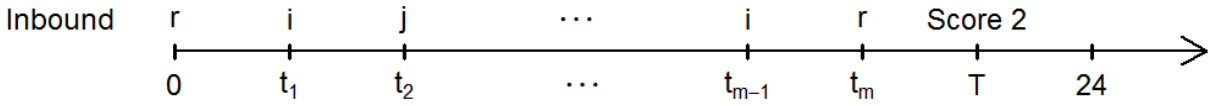


Figure 3.1: A basketball play. The ball is inbounded to player r at time 0; r passes the ball to i at time t_1; i passes the ball to j at time t_2; ...; player i receives the ball at time t_{m-1} and passes it to r at time t_m; the play ends when player r scores 2 points at time $T < 24$ seconds.

In a 48-minute NBA game, a team makes about 90-110 plays. Fewell et al. (2012) model basketball games as weighted networks by counting the frequencies of ball transitions

among the starts/ends of plays and the five positions of basketball players (point guard, shooting guard, small forward, power forward and center). Figure 3.2, which is taken from Fewell et al. (2012), displays the overall weighted network of 16 NBA games between 16 teams they have studied. The network illustrates play patterns and strategies on a game

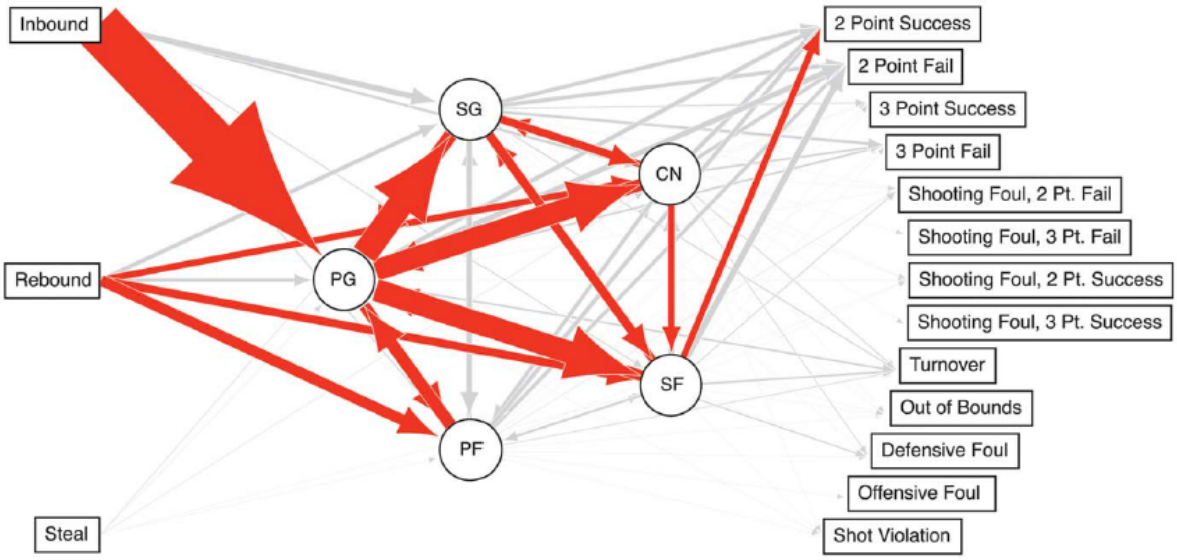


Figure 3.2: Weighted basketball network of 16 NBA games between 16 teams (Fewell et al., 2012). Circles represent the five positions (point guard, shooting guard, small forward, power forward, and center), and rectangles represent start or end points of a play. The width of the edge is proportional to the frequency of the corresponding ball transitions. The most frequent transition directions, which sum up to 60%, are colored red.

level. Fewell et al. (2012) compare the teams by investigating their networks. However, such a network can not capture any detail of real-time basketball play.

We explore basketball at the play level and take into account time effect. More specifically, we regard basketball as a transactional network. Table 3.1 illustrates our data. The two plays in the table are from game 1 of the 2012 NBA eastern conference finals between

the Miami Heat and the Boston Celtics. We manually collected the data by watching the videos of the games.

Table 3.1: Two plays from game 1 of the 2012 NBA eastern conference finals between the Boston Celtics and the Miami Heat. The top three lines show one play for the Boston Celtics. The ball is inbounded to C#9 (Rajon Rondo) at time 0; Rondo dribbles the ball and passes it to C#5 (Kevin Garnett) at second 11; Garnett misses a 2-pointer shot at second 12. Lines 4 to 9 illustrate one play for the Miami Heat.

From	**To**	**Time(s)**	**Players on the court**
Inbound	C#9	0	C#9, C#20, C#30, C#34
C#9	C#5	11	C#5, C#9, C#20, C#30, C#34
C#5	**Miss 2**	12	C#5, C#9, C#20, C#30, C#34
Rebound	H#6	0	H#3, H#6, H#15, H#21, H#31
H#6	H#3	7	H#3, H#6, H#15, H#21, H#31
H#3	H#15	8	H#3, H#6, H#15, H#21, H#31
H#15	H#3	9	H#3, H#6, H#15, H#21, H#31
H#3	H#6	12	H#3, H#6, H#15, H#21, H#31
H#6	**Miss** 3	17	H#3, H#6, H#15, H#21, H#31

In a basketball game, only ten players, five from each team, are on the court at one time. This means a basketball game is subject to many player substitutions. The last column of Table 3.1 records the players from the offensive team who are on the court at the events. Such information is necessary for our model. Note that the player inbounding the ball is treated as being off the court at the time of that event. For example, in Table 3.1, C#5 is inbounding the ball and not listed as being on the court.

As indicated earlier and shown in Figure 3.2, there are various ways to start and end a play. A play mostly starts with one of the three initial actions: inbounding, rebounding and stealing the ball. However, a play technically may end with about fifteen different

outcomes. For simplicity, we combine the outcomes to six categories: making a 2-pointer (Make 2), making a 3-pointer (Make 3), missing a 2-pointer (Miss 2), missing a 3-pointer (Miss 3), being fouled (Fouled) and making a turnover (TO). Scoring and being fouled at the same time is simply counted as scoring. Catching an air ball is counted as rebounding. All possible ways of giving up the possession of the ball such as direct turnover, being out of bound and offensive foul are regarded as turnover. We do not consider rare events such as a jump ball. We simply discard the rows corresponding to the rare events.

Although we group events into plays in Table 3.1, the model developed in the next section will treat each event as an individual occurrence, ignoring which play it belongs to. That is, the data in Table 3.1 will be seen as 9 isolated events (each with a timestamp), rather than 3 events in one play and 6 events in another play.

3.2 A Multistate CSBM for Basketball Networks

Our goal is to model the basketball network and cluster players into different groups, so that players in the same group have similar playing styles, while those in different groups play the game in more distinct ways. In this section, we propose a multistate Continuous-time Stochastic Block Model. The main idea is to adopt the Stochastic Block Model framework and model basketball plays as Continuous-time Markov Chains. An EM algorithm and a complementary algorithm are developed to fit the model. Although developed with basketball networks in mind, the model is applicable more broadly.

3.2.1 Pseudo Conditional Likelihood

During a basketball play (Figure 3.1), an initial action (e.g. inbounding) first transfers the ball to a player; the ball then moves among the players; finally, a play outcome is reached (e.g. the attacking team scores a 2-pointer). Hence, the ball moves among three types of nodes (see Section 3.1.2): a set of nodes $\mathcal{S} = \{\text{inbounding}, \text{rebounding}, \text{stealing}\}$ that designate different initial states, a total of n nodes that are players themselves, and a set of nodes $\mathcal{A} = \{\text{Make 2}, \text{Miss 2}, \text{Make 3}, \text{Miss 3}, \text{Fouled}, \text{TO}\}$ that designate different outcomes. In addition, we assume that there are K blocks, and each player only belongs to one block. The initial actions and the play outcomes are observable, but the blocks to which the players belong are not. Again, denote the block labels of the players by $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$, where $e_i \in \{1, 2, \ldots, K\}$. These block labels are latent. Following the conditional independence assumption of the SBM, the transactions among the nodes are independent given the block labels of the players. The conditional distribution for the entire basketball network, which includes all basketball plays, can be written as:

$$\mathcal{L}(\mathbf{T}|\mathbf{e}) = \left[\prod_{s\in\mathcal{S}}\prod_{i=1}^{n}\mathcal{L}^I(\mathbf{T}_{si}|\mathbf{e})\right]\cdot\left[\prod_{1\le i\neq j\le n}\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})\right]\cdot\left[\prod_{i=1}^{n}\prod_{a\in\mathcal{A}}\mathcal{L}^O(\mathbf{T}_{ia}|\mathbf{e})\right]. \quad (3.1)$$

where $\mathbf{T}_{si}$ denotes the transactions from an initial action s to player i; $\mathbf{T}_{ij}$ denotes the transactions from player i to player j; and $\mathbf{T}_{ia}$ denotes the transactions from player i to an outcome a. The conditional distribution (3.1) contains three natural components: $\mathcal{L}^I$, the distribution of all transactions from initial actions to players; $\mathcal{L}^P$, the distribution of

all passes among players; and $\mathcal{L}^O$, the distribution of all transactions from players to play outcomes. In the following subsections, we specify the details of these components one by one.

Transactions from initial actions to players

Define $\mathbf{P} = \{P_{sk} : s \in \mathcal{S}; k = 1, 2, \ldots, K\}$, where each P_{sk} is the probability that the basketball moves from initial action s to a player in block k. These probabilities are subject to the constraint that

$$\sum_{k=1}^{K} P_{sk} = 1, \text{ for any } s \in \mathcal{S}. \tag{3.2}$$

Given the block labels of all players, $\mathbf{e} = \{e_1, e_2, ..., e_n\}$, the distribution of the transactions from initial action s to player i is defined as

$$\mathcal{L}^I(\mathbf{T}_{si}|\mathbf{e}) = \prod_{h=1}^{m_{si}} \frac{P_{se_i} \cdot \frac{1}{G_{e_i}^{sih}}}{\sum_{k=1}^{K} \left(P_{sk} \cdot I(G_k^{sih} > 0)\right)}, \tag{3.3}$$

where m_{si} is the total number of times that a play goes from initial action s to player i. The quantity, G_k^{sih}, denotes the total number of "eligible receivers" belonging to block k for this particular play (from s to i), where "eligible receivers" are those players (including i here) who are on i's team and also physically on the basketball court (as opposed to sitting on the bench) at the h^{th} time that a transaction takes place from initial action s to player i. In general, we use the notation $G_k^{\triangle}$ to indicate the number of "eligible receivers" in block k at the time of an event indexed by $\triangle$. Quantities of this kind will appear a few

more times in the next few sections.

The definition (3.3) implies that players in the same cluster are stochastically equivalent. The probability that player i receives the ball from an initial action s is governed by the block-level probability P_{se_i} and individual-level probability $1/G_{e_i}^{sih}$, where we have assumed that all eligible receivers in the same cluster have an equal chance to receive the ball. The individual-level probability is needed in addition to the block-level probability because there is only one ball at all times and only one player can receive it. The denominator $\sum_{k=1}^{K}\left(P_{sk}\cdot I(G_k^{sih}>0)\right)$ is a normalization term, which takes into account the possible scenario that there may exist blocks without any eligible receivers on the court at the corresponding event. In such a scenario, the normalization term resales the transition probabilities, so that the overall probability that the ball goes from an initial action to an on-court player is equal to one.

Recall that we consider three initial actions: inbounding, rebounding and stealing. While rebounding and stealing both guarantee a new play, inbounding can start a new play or happen in the middle of a play. For example, a team may call a time-out in the middle of a play, and the play is resumed from the stoppage time by inbounding the ball. Another common situation is when an offensive player is fouled without being awarded free throws, the play is paused and resumed by inbounding the ball. We treat all inbounding events as initial actions and account for them in this part ($\mathcal{L}^I$) of the probability distribution.

Transactions among players

Intuitively, in a basketball play, what happens next mostly depends on the current situation, e.g., who has the ball at the moment, which players are on the court, and so on. Therefore, we model each basketball play as an inhomogeneous Markov chain. Players are treated as regular states; initial actions are treated as initial states; and play outcomes are modeled as absorbing states. We discussed transactions from initial states to regular states in Section 3.2.1. In this section, we focus on the regular states and construct $\prod_{1\leq i\neq j\leq n}\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})$ — the second component in (3.1), the conditional distribution of transactions among players, given the cluster labels $\mathbf{e}$.

Components of $\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})$ We now derive $\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})$, the conditional distribution of transactions from player i to j. To start, we revisit the basketball play shown in Figure 3.1 and isolate the segments related to the $i \to j$ process. For simplicity, suppose that player j is on the court during the entire play. As shown in Figure 3.3, player i first receives the ball at time t_1 and passes it to player j at time t_2, so the time period $(t_1, t_2]$ clearly belongs to the $i \to j$ process. Next, player i gains possession of the ball again at time t_{m-1} and the ball is passed to player $r \neq j$ at time t_m. Although player i does not make this pass to player j, he has the *potential* to do so. Hence, the time period (t_{m-1}, t_m) is also related to the $i \to j$ process. In fact, aside from the time point t_2 itself, there is no difference between the segments (t_{m-1}, t_m) and (t_1, t_2) in terms of being part of the $i \to j$ process — as long as i has possession of the ball, the segment is related to the $i \to j$ process, regardless of whether i actually passes the ball to j or not at the end of the segment. In Figure 3.3,

the segments related to the $i \to j$ process are highlighted by solid points and segments. Any solid point indicates an actual pass going from i to j. Any solid segment means that, during that time period, an i-to-j pass has the potential to happen.

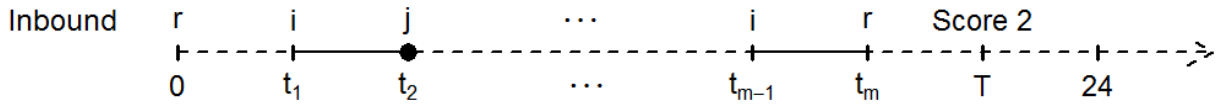


Figure 3.3: Segments of a play that are related to the $i \to j$ process. The $i \to j$ process consists of the solid point and the solid segments.

Note that only the segments when player i has the ball and player j is on the court are counted as part of the $i \to j$ process. In contrast, if j leaves the court in the middle of a play, anything happening afterward does not belong to the $i \to j$ process. According to NBA rules, player substitutions can only take place when certain event occurs and the game clock stops, meaning that no substitution can take place during any segment.

Given the cluster labels $\mathbf{e}$, we model each $i \to j$ process as pieces of a Poisson process. In addition, since each play is independent of one another, we can pool together all the "solid segments" and "solid points" (again, see Figure 3.3) from different plays. Again, we define K^2 rate functions, $\{\rho_{kl}(t) : k, l = 1, 2, \ldots, K\}$, where each $\rho_{kl}(t)$ is the rate that the ball moves from a player in cluster k to a player in cluster l at time t. By equation (2.1), the distribution of transactions from i to j is

$$\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e}) = \underbrace{\left[\prod_{h=1}^{m_{ij}} \left(\rho_{e_i e_j}(t_{ijh}) \cdot \frac{1}{G_{e_j}^{ijh}}\right)\right]}_{\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{e})} \cdot \underbrace{\left[\prod_{h=1}^{M_i} \exp\left(-\int_{t_{ih}^-}^{t_{ih}} \rho_{e_i e_j}(t) \cdot \frac{I_j^{ih}}{G_{e_j}^{ih}} \mathrm{d}t\right)\right]}_{\widetilde{\mathcal{L}}^{P_2}(\mathbf{T}_{ij}|\mathbf{e})}, \quad (3.4)$$

where

- m_{ij} is the total number of passes from i to j;
- t_{ijh} is the time of the h^{th} pass from i to j;
- $G_{e_j}^{ijh}$ is the number of "eligible receivers" belonging to block e_j for the h^{th} pass between i and j, with "eligible receivers" being those players (excluding i here) who are on i's team and also physically on the basketball court at the time of this pass;
- M_i is the total number of times that player i has possession of the ball;
- (t_{ih}^-, t_{ih}) is the h^{th} time interval in which player i has possession of the ball;
- $G_{e_j}^{ih}$ is the number of "eligible receivers" belonging to block e_j for the h^{th} pass from player i (regardless of whether j is the recipient or not); and
- the indicator I_j^{ih} is defined as

$$I_j^{ih} = \begin{cases} 1, & \text{if player } j \text{ is an "eligible receiver" for the } h^{th} \text{ pass from } i; \\ 0, & \text{otherwise.} \end{cases}$$

Note that the quantities, $G_{e_j}^{ih}$ and I_j^{ih}, are both constant on any interval $(t_{ih}^-, t_{ih}]$, since, as mentioned previously, the rules of the game prevent player substitutions during any such time interval. In addition, we have defined

$$\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{e}) \equiv \prod_{h=1}^{m_{ij}} \left(\rho_{e_i e_j}(t_{ijh}) \cdot \frac{1}{G_{e_j}^{ijh}} \right) \tag{3.5}$$

but written $\widetilde{\mathcal{L}}^{P_2}$ for the second component (rather than $\mathcal{L}^{P_2}$) because it can be simplified further (more details below) and this here is not the final expression we shall use.

In (3.4), the first term contains information about all passes from i to j, and the second term contains the information that i does not make a pass to j during all those time gaps in which i has possession of the ball and j, as a teammate of i, is on the court. The overall rate function for the $i \to j$ process consists of two distinctive parts. First, the rate function $\rho_{e_i e_j}(t)$ captures the rate of passing the ball at a cluster level. Second, similar to the fraction in (3.3), the fractions,

$$\frac{1}{G^{ijh}_{e_j}} \quad \text{and} \quad \frac{I^{ih}_j}{G^{ih}_{e_j}},$$

are the probabilities that player j is the actual receiver of the ball in group e_j. As in Section 3.2.1, we have assumed that all eligible receivers in the same cluster have an equal chance to receive the ball.

Notice that, if player j is off the court for a particular pass from i or if j is on the opponent team playing against i, then the fraction $I^{ih}_j/G^{ih}_{e_j}$ is automatically 0 by the definition of I^{ih}_j. In this way, time intervals (t^-_{ih}, t_{ih}) in which j is not an "eligible receiver" do not contribute to the $i \to j$ process, as one intuitively would expect. Furthermore, if $G^{ih}_{e_j} = 0$, it means there is no "eligible receiver" in block e_j — this can only happen if player j is not eligible itself, i.e., when $I^{ih}_j = 0$, because otherwise $G^{ih}_{e_j}$ is at least one since player j (always) belongs to block e_j. We define $0/0 = 0$. Finally, all time points, $\{t_{ijh} : i, j = 1, 2, \ldots, n; h = 1, 2, \ldots, m_{ij}\}$ and $\{t^-_{ih}, t_{ih} : i = 1, 2, \ldots, n; h = 1, 2, \ldots, M_i\}$, take values on the interval $[0, 24]$ (see Section 3.1.2).

Further simplification of $\widetilde{\mathcal{L}}^{P_2}$ So far, we have derived the (conditional) distribution of transactions from player i to player j, $\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})$. The conditional independence assumption means the (conditional) distribution of transactions between all pairs of players is simply

$$\prod_{1\le i\neq j\le n}\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e}) = \left[\prod_{1\le i\neq j\le n}\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{e})\right]\cdot\left[\prod_{1\le i\neq j\le n}\widetilde{\mathcal{L}}^{P_2}(\mathbf{T}_{ij}|\mathbf{e})\right].$$

The second term above can be simplified further. In particular,

$$\begin{aligned}
\prod_{1\le i\neq j\le n}\widetilde{\mathcal{L}}^{P_2}(\mathbf{T}_{ij}|\mathbf{e}) &= \prod_{1\le i\neq j\le n}\prod_{h=1}^{M_i}\exp\Big(-\int_{t_{ih}^-}^{t_{ih}}\rho_{e_ie_j}(t)\cdot\frac{I_j^{ih}}{G_{e_j}^{ih}}\mathrm{d}t\Big) \qquad (3.6)\\
&= \prod_{i=1}^{n}\prod_{h=1}^{M_i}\prod_{j\neq i}\exp\Big(-\int_{t_{ih}^-}^{t_{ih}}\rho_{e_ie_j}(t)\cdot\frac{I_j^{ih}}{G_{e_j}^{ih}}\mathrm{d}t\Big)\\
&= \prod_{i=1}^{n}\prod_{h=1}^{M_i}\exp\left[-\int_{t_{ih}^-}^{t_{ih}}\sum_{j\neq i}\left(\rho_{e_ie_j}(t)\cdot\frac{I_j^{ih}}{G_{e_j}^{ih}}\right)\mathrm{d}t\right]\\
&= \prod_{i=1}^{n}\prod_{h=1}^{M_i}\exp\left[-\int_{t_{ih}^-}^{t_{ih}}\sum_{l=1}^{K}\sum_{\substack{j\neq i\\ e_j=l}}\left(\rho_{e_il}(t)\cdot\frac{I_j^{ih}}{G_l^{ih}}\right)\mathrm{d}t\right]\\
&= \prod_{i=1}^{n}\prod_{h=1}^{M_i}\exp\left[-\int_{t_{ih}^-}^{t_{ih}}\sum_{l=1}^{K}\left(\rho_{e_il}(t)\cdot\sum_{\substack{j\neq i\\ e_j=l}}\frac{I_j^{ih}}{G_l^{ih}}\right)\mathrm{d}t\right].
\end{aligned}$$

Notice that, on the set $e_j = l$, whenever $G_l^{ih} = 0$ (i.e., nobody in block l is an eligible receiver), we must have $I_j^{ih} = 0$ as well (i.e., player j cannot be an eligible receiver, either,

since $e_j = l$ means player j belongs to block l). Therefore,

$$\sum_{\substack{j \neq i \\ e_j = l}} \frac{I_j^{ih}}{G_l^{ih}} = I(G_l^{ih} > 0).$$

Continuing with (3.6), this means

$$\prod_{1 \leq i \neq j \leq n} \widetilde{\mathcal{L}}^{P_2}(\mathbf{T}_{ij}|\mathbf{e}) = \underbrace{\prod_{i=1}^{n} \prod_{h=1}^{M_i} \exp\left[- \int_{t_{ih}^-}^{t_{ih}} \sum_{l=1}^{K} \Big(\rho_{e_i l}(t) \cdot I(G_l^{ih} > 0) \Big) \mathrm{d}t \right]}_{\mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{e})}. \quad (3.7)$$

Decomposition of $\mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e})$ Putting all the pieces together, the conditional distribution of all transactions among players, given the block labels, is of the form

$$\prod_{1 \leq i \neq j \leq n} \mathcal{L}^P(\mathbf{T}_{ij}|\mathbf{e}) = \left[\prod_{1 \leq i \neq j \leq n} \mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{e}) \right] \cdot \left[\prod_{i=1}^{n} \mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{e}) \right]. \quad (3.8)$$

The first component, $\prod_{i \neq j} \mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{e})$, contains information about all passes from i to j. The second component, $\prod_{i=1}^{n} \mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{e})$, contains information about all the time gaps in which player i has possession of the ball — although, admittedly, denoting all these time gaps here by $\mathbf{T}_i$ is a slight abuse of notation.

In equation (3.7), the indicator $I(G_l^{ih} > 0)$ is important for two reasons. First, if node i is the only member in group l or if group l is empty, then it is impossible for i to pass the ball to group l, so intuitively the rate function $\rho_{e_i l}(t)$ should not contribute any information to

this part of the probability distribution. Indeed, in either situation, we have $G_l^{ih} = 0$, and this indicator effectively "annihilates" the contribution of $\rho_{e_i l}$. Second, we can see from (3.7) that, overall, player i has a rate of $\sum_{l=1}^{K} \left(\rho_{e_i l}(t) \cdot I(G_l^{ih} > 0)\right)$ to pass the ball at time t. Given $\rho_{kl}(t)$, when there are fewer groups for player i to pass the ball to, its overall rate of passing the ball is automatically reduced by this indicator, which agrees with our intuition about how basketball games are played.

Notice that in (3.4), because of the existence of $1/G_{e_j}^{ijh}$, the rate function from player i to player j actually also depends on the cluster labels of the other on-court players. Hence, the likelihood is a pseudo likelihood (Besag, 1975).

Transactions from players to play outcomes

The play outcomes are modeled as absorbing states of the Markov chain. Given a set $\mathcal{A}$ of different play outcomes, we define additional rate functions $\{\eta_{ka}(t) : k = 1, 2, \ldots, K; a \in \mathcal{A}\}$, where $\eta_{ka}(t)$ is the rate that a play goes from group k to absorbing state a at time t.

Whenever player i has possession of the ball, there exists a possibility that the ball is "passed" to an absorbing state, a. Analogous to (3.4), the distribution of transactions from player i to an absorbing state a can be written as

$$\mathcal{L}^O(\mathbf{T}_{ia}|\mathbf{e}) = \left[\prod_{h=1}^{m_{ia}} \eta_{e_i a}(t_{iah})\right] \cdot \left[\prod_{h=1}^{M_i} \exp\Big(-\int_{t_{ih}^-}^{t_{ih}} \eta_{e_i a}(t)\mathrm{d}t\Big)\right], \tag{3.9}$$

where m_{ia} is the total number of times that the ball goes from node i to absorbing state a;

and t_{iah} is the time of the h^{th} event from i to a — except that we need no longer multiply the rate function $\eta_{e_i a}(\cdot)$ by an additional individual-level probability (such as $1/G_{e_i}^{iah}$), since there aren't multiple options within an absorbing state as there can be multiple players in a cluster. As in (3.4), the first term contains information about the event times, and the second term contains the information that player i does not "cause" the play to end in absorbing state a while in possession of the ball.

Even though being fouled does not always end a play, we still consider being fouled as an "outcome" and take account of all fouls in this part ($\mathcal{L}^O$) of the probability distribution.

A Markov chain

Here is a brief recapitulation of how we have modeled basketball networks (Sections 3.2.1, 3.2.1 and 3.2.1) conditional on the cluster labels of the players. There are three types of nodes in the network: *special* nodes that designate initial actions, *regular* nodes that are players themselves, and *terminal* nodes that designate play outcomes. If we isolate any two regular nodes, or a regular node and a terminal node, transactions between those two nodes have been modelled as an inhomogeneous Poisson process. Each basketball play, however, will consist of a sequence of transactions — typically starting from a special node, travelling across multiple regular nodes, and ending in a terminal node. Each play is thus an inhomogeneous, continuous-time Markov chain, of which the players are regular states and outcomes are absorbing states.

Nonparametric modeling of rate functions

Again, we model the rate functions nonparametrically by cubic B-splines:

$$\rho_{kl}(t) = \sum_{p=1}^{P} e^{\beta_{klp}} B_p(t), \text{ for } k, l = 1, 2, \ldots, K, \tag{3.10}$$

$$\eta_{ka}(t) = \sum_{p=1}^{P} e^{\psi_{kap}} B_p(t), \text{ for } k = 1, 2, \ldots, K \text{ and } a \in \mathcal{A}, \tag{3.11}$$

where $\{B_1(t), B_2(t), \ldots, B_P(t)\}$ are basis functions; and $\boldsymbol{\beta} = \{\beta_{klp} : k, l = 1, 2, \ldots, K; p = 1, 2, \ldots, P\}$, $\boldsymbol{\psi} = \{\psi_{kap} : k = 1, 2, \ldots, K; a \in \mathcal{A}; p = 1, 2, \ldots, P\}$ are coefficients. We use exponentiated coefficients, $e^{\beta_{klp}}$ and $e^{\psi_{kap}}$, to ensure that all rate functions are nonnegative.

Remarks

A cluster can contain players from different teams, although players from different teams are not able to pass the ball to each other. The multistate CSBM clusters players according to their playing styles, and it is very likely that different teams have similar players in terms of playing styles. For instance, suppose player i_A from team A and player i_B from team B both belong to cluster k, and player j_A from team A and player j_B from team B both belong to cluster j, then the passing rate from i_A to j_A at time t is $\rho_{kl}(t)$, and so is the passing rate from i_B to j_B. The fact that i_A can not make a pass to j_B does not affect the estimate of $\rho_{kl}(t)$, because the model incorporates indicators for "eligible receivers" of the ball, i.e., I_j^{ih} and $G_{e_j}^{ijh}$.

3.2.2 An EM$^+$ Algorithm

In the same manner as for the basic CSBM in Chapter 2, we introduce latent variables and adopt the Expectation-Maximization (EM) algorithm to fit the multistate CSBM. As discussed in 2.4, the EM algorithm converges quickly because it essentially behaves like the k-means algorithm. It works well for the simple simulation example in Section 2.3. However, the multistate CSBM is more complex. We have found in our experience that the EM algorithm alone can sometimes be trapped in various local optima. Running the EM algorithm with many random starting points helps, but it is quite inefficient. Instead, we have added a complementary heuristic algorithm to run *after* the EM algorithm. We refer to the complementary algorithm as the "Plus algorithm" and call our overall algorithm an "EM$^+$ algorithm". Empirically, we have found that the EM$^+$ algorithm often reaches a nice optimal point with fewer starting points than does the EM algorithm itself.

EM Algorithm

Let $\boldsymbol{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iK})$ denote a latent label indicator for node i, such that

$$z_{ik} = \begin{cases} 1, & \text{if node } i \text{ belongs to cluster } k; \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

Marginally,

$$\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_n \overset{iid}{\sim} \text{multinomial}(1, \boldsymbol{\pi}), \quad \text{where} \quad \boldsymbol{\pi} = (\pi_1, \ldots, \pi_K).$$

We shall use $\Theta = \{\mathbf{P}, \boldsymbol{\beta}, \boldsymbol{\psi}, \boldsymbol{\pi}\}$ to denote all parameters, and $\mathbf{Z} = \{\boldsymbol{z_i} : i = 1, 2, \ldots, n\}$ to denote all latent indicators. The complete likelihood of the multistate CSBM is simply the joint distribution of $(\mathbf{T}, \mathbf{Z})$ viewed as a function of Θ. To simplify our notation as well as to make more direct references to the models we described in Section 3.2.1, in this section we will often suppress Θ and still write $\mathcal{L}(\mathbf{T}, \mathbf{Z})$ instead of $\mathcal{L}(\Theta; \mathbf{T}, \mathbf{Z})$ for the likelihood function. Hence, the complete likelihood is

$$\mathcal{L}(\mathbf{T}, \mathbf{Z}) = \mathcal{L}(\mathbf{T}|\mathbf{Z}) \cdot \mathcal{L}(\mathbf{Z}). \tag{3.13}$$

The conditional likelihood $\mathcal{L}(\mathbf{T}|\mathbf{Z})$ is simply a latent-variable-coded version of $\mathcal{L}(\mathbf{T}|\mathbf{e})$ (3.1), that is,

$$\begin{aligned}
&\mathcal{L}(\mathbf{T}|\mathbf{Z}) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.14)\\
&= \Big[\prod_{s\in\mathcal{S}}\prod_{i=1}^{n}\mathcal{L}^{I}(\mathbf{T}_{si}|\mathbf{Z})\Big] \cdot \Big[\prod_{1\le i\neq j\le n}\mathcal{L}^{P}(\mathbf{T}_{ij}|\mathbf{Z})\Big] \cdot \Big[\prod_{i=1}^{n}\prod_{a\in\mathcal{A}}\mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z})\Big]\\
&= \Big[\prod_{s\in\mathcal{S}}\prod_{i=1}^{n}\mathcal{L}^{I}(\mathbf{T}_{si}|\mathbf{Z})\Big] \cdot \Big[\prod_{1\le i\neq j\le n}\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{Z}) \cdot \prod_{i=1}^{n}\mathcal{L}^{P_2}(\mathbf{T}_{i}|\mathbf{Z})\Big]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \Big[\prod_{i=1}^{n}\prod_{a\in\mathcal{A}}\mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z})\Big],
\end{aligned}$$

where the second step above is due to (3.8). More specifically, the components of (3.14) are simply latent-variable versions of (3.3), (3.5), (3.7) and (3.9):

$$\mathcal{L}^{I}(\mathbf{T}_{si}|\mathbf{Z}) \;=\; \prod_{k=1}^{K}\Big[\prod_{h=1}^{m_{si}}\big(P_{sk}\cdot\frac{1}{G_k^{sih}}\big)\Big]^{z_{ik}}, \tag{3.15}$$

$$\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{Z}) = \prod_{k=1}^{K}\prod_{l=1}^{K}\Big[\prod_{h=1}^{m_{ij}}\Big(\rho_{kl}(t_{ijh})\cdot\frac{1}{G_l^{ijh}}\Big)\Big]^{z_{ik}z_{jl}}, \tag{3.16}$$

$$\mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{Z}) = \prod_{k=1}^{K}\Big[\prod_{h=1}^{M_i}\exp\Big(-\sum_{l=1}^{K}\int_{t_{ih}^-}^{t_{ih}}\rho_{kl}(t)\cdot I(G_l^{ih}>0)\mathrm{d}t\Big)\Big]^{z_{ik}}, \tag{3.17}$$

$$\mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z}) = \prod_{k=1}^{K}\Big[\prod_{h=1}^{m_{ia}}\eta_{ka}(t_{iah})\cdot\prod_{h=1}^{M_i}\exp\Big(-\int_{t_{ih}^-}^{t_{ih}}\eta_{ka}(t)\mathrm{d}t\Big)\Big]^{z_{ik}}. \tag{3.18}$$

The marginal likelihood of $\mathbf{Z}$ is

$$\mathcal{L}(\mathbf{Z}) = \prod_{i=1}^{n}\prod_{k=1}^{K}\pi_k^{z_{ik}}. \tag{3.19}$$

E-step In the E-step, we compute $\mathbf{E}\big(\log\mathcal{L}(\mathbf{T},\mathbf{Z})|\mathbf{T};\Theta^*\big)$, the conditional expectation of the log-likelihood given the observed network $\mathbf{T}$ under the current parameter estimates (denoted by Θ^*). The conditional expectation is with respect to the latent variables $\mathbf{Z}$. From (3.15)-(3.18) it is clear (details in the Appendix A.1) that now there are three types of conditional expectations to evaluate:

- $\mathbf{E}(z_{ik}|\mathbf{T};\Theta^*)$, from $\log\mathcal{L}^I(\mathbf{T}_{si}|\mathbf{Z})$, $\log\mathcal{L}^O(\mathbf{T}_{ia}|\mathbf{Z})$ and $\log\mathcal{L}(\mathbf{Z})$, respectively;
- $\mathbf{E}(z_{ik}z_{jl}|\mathbf{T};\Theta^*)$, from $\log\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{Z})$; and
- $\mathbf{E}\big(z_{ik}\cdot I(G_l^{ih}>0)|\mathbf{T};\Theta^*\big)$, from $\log\mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{Z})$.

After taking logarithms, the terms involving $1/G_k^{sih}$ and $1/G_l^{ijh}$ in (3.15) and (3.16) are additive "constants" that depend only on the latent variables $\mathbf{Z}$ but contain no information

about the parameters Θ; they can be omitted for the EM algorithm. The quantity

$$G_l^{ih} = \sum_{j \neq i} \left(z_{jl} \cdot I_j^{ih} \right)$$

and hence the indicator $I(G_l^{ih} > 0)$ are both functions of the latent variables. Here, we see more clearly why the further simplification of $\mathcal{L}^{P_2}$ — equation (3.7) — is useful. As in Section 2.2.2, due to the interactions of the players, the latent variables are conditionally dependent and an exact calculation of the conditional expectations above is NP-hard. Again, we use a Gibbs sampler to draw samples from $\mathcal{L}(\mathbf{Z}|\mathbf{T}; \Theta^*)$, and use the corresponding sample means to approximate $\mathbf{E}(z_{ik}|\mathbf{T}; \Theta^*)$, $\mathbf{E}(z_{ik} z_{jl}|\mathbf{T}; \Theta^*)$ and $\mathbf{E}\big(z_{ik} \cdot I(G_l^{ih} > 0)|\mathbf{T}; \Theta^*\big)$. The Gibbs sampler is the same as the one in Section 2.2.2 except that the distribution functions are updated.

M-step In the M-step, we update the parameters Θ by maximizing $\mathbf{E}\big(\log \mathcal{L}(\mathbf{T}, \mathbf{Z})|\mathbf{T}; \Theta^*\big)$. Same as in Section 2.2.2, we have closed-form solutions for $\boldsymbol{\pi}$, the marginal probabilities of $\mathbf{Z}$:

$$\pi_k = \frac{\sum_{i=1}^n \mathbf{E}(z_{ik}|\mathbf{T}; \Theta^*)}{\sum_{i=1}^n \sum_{l=1}^K \mathbf{E}(z_{il}|\mathbf{T}; \Theta^*)} = \frac{\sum_{i=1}^n \mathbf{E}(z_{ik}|\mathbf{T}; \Theta^*)}{n}, \tag{3.20}$$

for $k = 1, 2, \ldots, K$. Once more, there are no closed-form solutions for $\boldsymbol{\beta}$ and $\boldsymbol{\psi}$, the (log)-coefficients for the rate functions. Following the same approach described in Section 2.2.2, we use the quasi-Newton method with L-BFGS-B updates. Regarding $\mathbf{P}$, the transition probabilities from initial states, the exact maximum likelihood estimations require numer-

ical approaches because of the normalization terms in (3.3). We can use the `constrOptim` function in R.

Remarks For the basic CSBM, we have discussed in Section 2.4 that the conditional probabilities driving the Gibbs sampler are fairly close to 0 or 1, and the Gibbs sampler thus converges very quickly to a singular probability mass. Hence, the EM algorithm is reduced to a k-means algorithm and converges in just a few iterations. With the complexity of the multistate CSBM, the EM algorithm can sometimes be trapped in a local optimum. The typical way to avoid these traps is to use different starting points, run the EM algorithm for a few times, and pick the one giving the largest likelihood value. This "standard" procedure alone could be quite inefficient. Instead, we introduce another heuristic algorithm, which we refer to as the Plus algorithm, as a complement to the EM algorithm. Sometimes, e.g., when the EM solution is already quite good, the Plus algorithm may not find any further improvement.

The Plus Algorithm

This algorithm is inspired by the heuristic algorithm used by Karrer and Newman (2011) for the so-called degree-corrected SBM. The main idea is to evaluate all neighbors of the current labelling configuration and move to the best neighbor no matter whether the likelihood improves or not. A neighbor of a labelling configuration $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ is defined as the one with only one entry being different. Thus, if $\mathbf{e}'$ and $\mathbf{e}$ are neighbors, then there exists some $1 \leq i \leq n$ such that $e_i \neq e_i'$, but otherwise $e_j = e_j'$ for all $j \neq i$.

Given n nodes and K clusters, one labelling configuration has $n(K-1)$ neighbors. The steps of the algorithm are as follows.

1. Start with $r = 0$.
2. Repeat the following steps until convergence, or for a fixed number of steps.
   (a) Given a labelling configuration $\mathbf{e}^{(r)}$ and parameter $\Theta^{(r)}$ estimated under $\mathbf{e}^{(r)}$, calculate the likelihood of all neighboring configurations, using the same parameter estimate, $\Theta^{(r)}$.
   (b) Let $\mathbf{e}^{(r+1)}$ be the neighbor that gives the largest likelihood.
   (c) Re-estimate the parameters using $\mathbf{e}^{(r+1)}$, and denote the result by $\Theta^{(r+1)}$.
3. Choose the best configuration among $\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \mathbf{e}^{(2)}, ...$

We use the result from the EM algorithm as the starting point $\mathbf{e}^{(0)}$ to run the Plus algorithm. The Plus algorithm converges when there exists a set of configurations $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_q$ such that $\mathbf{e}_1$ is the best neighbor of $\mathbf{e}_2$, $\mathbf{e}_2$ is the best neighbor of $\mathbf{e}_3$, ..., and $\mathbf{e}_q$ is the best neighbor of $\mathbf{e}_1$. Often, this happens for $q = 2$, but sometimes it can happen for $q > 2$. Note that, while $\mathbf{e}^{(r+1)}$ in step (2b) gives the largest likelihood among all neighbors of $\mathbf{e}^{(r)}$, it may still give a smaller likelihood than does $\mathbf{e}^{(r)}$ itself, but the Plus algorithm "accepts" $\mathbf{e}^{(r+1)}$ nonetheless. This is the main reason why the Plus algorithm can help the EM algorithm avoid local optima. On the other hand, the Plus algorithm itself moves very slowly — in any given iteration, only one node label is changed, so it is quite inefficient to use it as a standalone algorithm, but we have found it to work well as a complement to the EM algorithm.

3.3 Applications to NBA data

In this section, we apply the multistate Continuous-time Stochastic Block Model to a few NBA basketball games that we have annotated ourselves. The games are: the 2012 NBA eastern conference finals between the Miami Heat and the Boston Celtics, games 1 and 5; and the 2015 NBA finals between the Cleveland Cavaliers and the Golden State Warriors, games 2 and 5. For each game, we only consider the first three quarters to avoid having to deal with garbage time or irregular playing strategies (such as committing fouls on purpose), which are both common in the last quarter. In Section 3.3.1, we present some further model simplifications and corresponding adjustments to the EM$^+$ algorithm. In Sections 3.3.2 and 3.3.3, we present results for 2012 games between the Heat and the Celtics, and those for the 2015 games between the Cavaliers and the Warriors, respectively. In Section 3.3.4, we compare the 2012 Miami Heat with the 2015 Cleveland Cavaliers, while paying special attention to the performance of LeBron James as he played with these two different teams in those two series.

3.3.1 Model simplifications and adjustments of the EM$^+$ algorithm

In practice, the general model is complex, with $K(K+|\mathcal{A}|)$ rate functions to estimate. For applications to NBA data, we further simplify the general form by defining

$$\rho_{kl}(t) = \lambda_k(t) \cdot P_{kl}, \tag{3.21}$$

$$\eta_{ka}(t) = \lambda_k(t) \cdot P_{ka}, \tag{3.22}$$

such that $\lambda_k(t)$ is the rate function of the ball leaving a player in group k; P_{kl} and P_{ka} are transition probabilities that the ball goes to group l and absorbing state a, respectively. The transition probabilities are subject to the constraint

$$\sum_{l=1}^{K} P_{kl} + \sum_{a\in\mathcal{A}} P_{ka} = 1, \text{for any } k = 1, 2, \ldots, K. \tag{3.23}$$

By making such simplifications, we assume that, whenever the ball leaves cluster k, the rates to other clusters and absorbing states are formed by a common rate and proportionality constants. In reality, the transition probabilities may change over time, but we believe that the simplified model still contains sufficient information to cluster players and reveal important patterns. The results in the next section provide convincing evidence.

The rate function simplifications lead to modifications in the EM$^+$ algorithm. Now the $K(K+|\mathcal{A}|)$ rate functions reduce to K rate functions and a $K\times(K+|\mathcal{A}|)$ transition matrix. We still adopt quasi-Newton for the rate functions, yet we have closed-form solutions for the transition probabilities (details in Appendix A.3),

$$P_{kl} = \frac{\sum_{1\le i\ne j\le n}\Big(\mathbf{E}[z_{ik}z_{jl}|\mathbf{T};\Theta^*]\cdot m_{ij}\Big)}{\sum_{i=1}^{n}\sum_{h=1}^{M_i}\Big(\mathbf{E}\Big[z_{ik}I(G_l^{-i}(t_{ih})>0)\Big|\mathbf{T};\Theta^*\Big]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\Big)+\zeta_k}, \tag{3.24}$$

$$P_{ka} = \frac{\sum_{i=1}^{n}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{ia}\Big)}{\sum_{i=1}^{n}\sum_{h=1}^{M_i}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\Big)+\zeta_k}, \tag{3.25}$$

for $k, l = 1, 2, \ldots, K$ and $a \in \mathcal{A}$. The parameter ζ_k is the Lagrange multiplier, which can be easily solved by finding the root of $\sum_{l=1}^{K} P_{kl} + \sum_{a\in\mathcal{A}} P_{ka} = 1$ with the R function

`uniroot`.

The marginal probabilities remain unchanged, so the update equation (3.20) still applies. For the initial probabilities $\{P_{sk} : s \in \mathcal{S}; k = 1, 2, \ldots, K\}$, instead of adopting numerical methods for exact estimations, we use some approximations to save computing time. In our applications, we pick $K = 3$ or 4, which are small, so all clusters have players on the court in most of the time. This fact implies that almost all normalization terms in (3.3), i.e., $\sum_{k=1}^{K} \left(P_{sk} \cdot I(G_k^{sih} > 0)\right)$ for all s, i and h, are equal to one. Hence, we can simply approximate (3.3) by ignoring the normalization terms. Such an approximation yields closed-form solutions for the initial probabilities:

$$P_{sk} = \frac{\sum_{i=1}^{n} [m_{si} \mathbf{E}(z_{ik} | \mathbf{T}; \Theta^*)]}{\sum_{k=1}^{K} \sum_{i=1}^{n} [m_{si} \mathbf{E}(z_{ik} | \mathbf{T}; \Theta^*)]}, \tag{3.26}$$

for $s \in \mathcal{S}$ and $k = 1, 2, \ldots, K$; detailed derivations are given in Appendix A.2.

For this simplified model, all probability parameters including marginal probabilities π_k, initial probabilities P_{sk} and transition probabilities P_{kl} and P_{ka} have closed-from updates. Hence, to make the EM$^+$ algorithm more efficient, we partition the parameter set Θ into two groups: $\Theta_{fast} = \{\pi_k, P_{sk}, P_{kl}, P_{ka}\}$, consisting of all parameters with closed-form updates, and $\Theta_{slow} = \{\lambda_k(t)\}$, consisting of all parameters that we must update with quasi-Newton. Instead of updating all Θ only in the M-step of the EM algorithm and Step (2c) of the Plus algorithm, parameters belonging to Θ_{fast} are always updated instantaneously "on the fly" — meaning that they are updated whenever there is a change in $\mathbf{Z}$ or the cluster labels $\mathbf{e}$. More specifically, Θ_{fast} are updated when calculating each likelihood function in the Gibbs sampler of the EM algorithm and Step (2a) of the plus algorithm.

We use 15 cubic B-spline basis functions. The EM algorithm is run multiple times from different random starting points. More explicitly, in the first E-step of each run, the starting value for label configuration is randomized. For our applications, we find that running the EM^+ algorithm for 2 – 5 times is enough to yield the "best" results.

3.3.2 Miami Heat versus Boston Celtics in 2012

In the 2012 NBA eastern conference finals, eleven players from the Heat and ten players from the Celtics played in the first three quarters of their 1st and 5th games. We omit two Celtics players, Ryan Hollins and Marquis Daniels, because they each touched the ball only once in those quarters. The data, which have been illustrated in Table 3.1, consist of 283 plays (142 for the Heat and 141 for the Celtics) and 1205 transactions (657 for the Heat and 548 for the Celtics). We fit three different CSBMs — one to the Heat's transactions alone, one to the Celtics' transactions alone, and another one to transactions from both teams pooled together. In what follows, we discuss in detail our clustering results, initial probability estimates, fitted rate functions, and transition probability estimates. Given our data size (11 Heat players and 8 Celtics players), we picked a moderate number of clusters ($K = 3$). In practice, since the main purpose of our model is to cluster players and narrow down the search space for basketball scouts, the choice of K will mostly depend on the size of the basketball network and how elaborate one wants the clustering results to be.

Clustering results The cluster labels for the players are reported in Table 3.2. Recall that basketball players play in five different positions: point guard (PG), shooting guard

(SG), small forward (SF), power forward (PF) and center (C). Generally speaking, the heights of the players are PG<SG<SF<PF<C.

Considered separately, players in the two teams are clustered in similar manners. Point guards are in cluster 1; two perimeter players — {Wade, James} from the Heat and {Allen, Pierce} from the Celtics — are in cluster 2; and the other players are in cluster 3. Roughly speaking, players with similar heights and close positions are clustered into the same group. Point guards certainly play in a different style than those of power forwards and centers. Shooting guards and small forwards are both perimeter players and often play in similar styles. In our case, Wade, James, Allen and Pierce are different than the other perimeter players, because they are stars. They have extraordinary offensive skills, so they can carry the ball longer and shoot more often. By contrast, the shooting guards and small forwards in cluster 3 play without the ball for most of the time.

When the two teams are pooled together, only one player (Brandon Bass from the Celtics) switches from cluster 3 to cluster 2. Actually, he is a "mini" PF, who has a typical PF's weight and strength but the height of an SF, so his playing style is in between those of a typical SF and a typical PF. When compared only with other Celtics players, he is more similar to those in cluster 3. However, when players from the Heat also are included in the comparison, he starts to look more similar to LeBron James (a strong SF) and very different than those in cluster 3 who are on the Heat, e.g., in terms of rebounding, cutting, post playing, so he is re-clustered into cluster 2.

In our subjective assessment, players in cluster 1 tend to dribble the ball a lot but do not shoot very often, those in cluster 2 both carry and shoot the ball, whereas those in cluster

Table 3.2: Clustering results for the 2011-2012 Miami Heat and Boston Celtics ($K = 3$). Cluster labels are C1, C2, C3. Three different clustering results are presented (two under "Alone" and one under "Together"). Player positions are included for reference only; they are not used by the clustering algorithm.

			Alone			Together		
Team	Player	Position	C1	C2	C3	C1	C2	C3
Heat	Mario Chalmers	PG	X			X		
	Norris Cole	PG	X			X		
	Dwyane Wade	SG		X			X	
	LeBron James	SF		X			X	
	James Jones	SG			X			X
	Shane Battier	SF			X			X
	Mike Miller	SF			X			X
	Chris Bosh	PF			X			X
	Udonis Haslem	PF			X			X
	Ronny Turiaf	C			X			X
	Joel Anthony	C			X			X
Celtics	Rajon Rondo	PG	X			X		
	Keyon Dooling	PG	X			X		
	Ray Allen	SG		X			X	
	Paul Pierce	SF		X			X	
	Mickael Pietrus	SF			X			X
	Brandon Bass	PF			X		X	
	Kevin Garnett	C			X			X
	Greg Stiemsma	C			X			X

Table 3.3: Estimated transition probabilities (P_{sk}) from each initial action to clusters C1, C2, C3, for three different clustering models of the 2011-2012 Miami Heat and Boston Celtics.

		C1	C2	C3
Heat	Inbound	0.716	0.194	0.090
	Rebound	0.109	0.375	0.516
	Steal	0.333	0.333	0.333
Celtics	Inbound	0.868	0.059	0.073
	Rebound	0.188	0.208	0.604
	Steal	0.375	0.500	0.125
Together	Inbound	0.793	0.133	0.074
	Rebound	0.143	0.357	0.500
	Steal	0.364	0.454	0.182

3 are mostly responsible for catching rebounds and shooting, but not so much for carrying the ball. In what follows, we will see these differences of the three clusters reflected in the different parameters of the CSBM.

Initial probabilities Table 3.3 displays the estimated transition probabilities from each initial action to the three clusters. Most inbounds go to point guards, because they usually are the ones to carry the ball from the back court to the front court. The Heat inbound more often to cluster 2 than the Celtics do, because LeBron James (in cluster 2) sometimes plays like a point guard. More than half of the rebounds are caught by cluster 3, the tall players. For the Celtics, their cluster 1 players catch almost as many rebounds as those in their cluster 2, because the starting point guard, Rajon Rondo (in cluster 1), is an excellent rebounder. Regarding steals (a relatively rare event), the three clusters contribute equally within the Heat but somewhat differently within the Celtics.

Rate functions Figure 3.4 contains the fitted rate functions $\{\lambda_k(t) : k = 1, 2, 3\}$ for the ball leaving a player in group k. Overall, these functions are quite different for the three clusters. For the same cluster, the rate functions from different teams are similar in general, but have considerable differences at certain time points. Below, we compare the patterns of the rate functions over four distinct time periods: $t \in (0, 5)$, $t \in (5, 10)$, $t \in (10, 15)$, and $t > 15$.

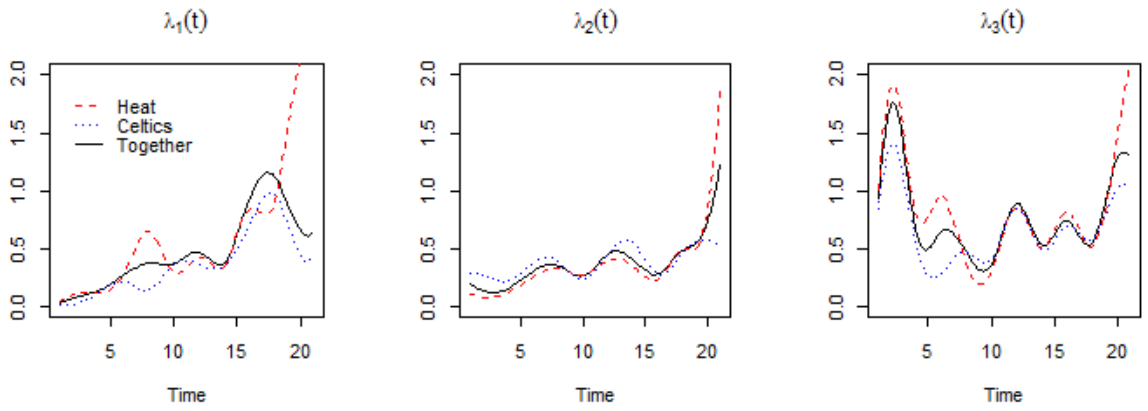


Figure 3.4: Fitted rate functions for the 2011-2012 Miami Heat and Boston Celtics, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rate with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively.

At the beginning of a play, it usually takes about five seconds for a point guard to dribble the ball from the back court to the front court. Players in cluster 2 sometimes do that instead of point guards. Therefore, $\lambda_1(t)$ and $\lambda_2(t)$ are low for $t \in (0, 5)$. However, for both teams their $\lambda_3(t)$ has a high and sharp peak around $t \approx 2$, because players in cluster 3 often catch rebounds and start new plays by quickly passing the ball to those in the other two clusters.

After the ball arrives at the front court, the players spend about 5 seconds to settle down to their offensive layout. During this time period, i.e., $t \in (5, 10)$, the two teams have different strategies. For the Heat, the point guards usually pass the ball to either James or Wade and let them handle the ball, so we can see a small peak in the Heat's $\lambda_1(t)$ function. For the Celtics, their point guards — especially Rondo — usually continue to hold the ball and organize the offense, so the Celtics' $\lambda_1(t)$ function even declines a little right after $t > 5$. The two teams' $\lambda_3(t)$ functions exhibit significant difference over this time period. For the Heat, their players in cluster 3 mostly play as transit ports, i.e., they get the ball and pass it out soon. For the Celtics, their players in cluster 3 — especially Kevin Garnett — have more opportunities to handle the ball. That is why in the right panel, the Heat's $\lambda_3(t)$ function has a peak around $t \approx 7$, while the Celtics' $\lambda_3(t)$ has a local minimum between $5 < t < 6$.

For $t \in (10, 15)$, if the play still keeps going, players start to pass the ball more frequently and seek scoring opportunities. This is indicated by higher values in $\lambda_1(t)$ and $\lambda_2(t)$ as well as a local peak in $\lambda_3(t)$ on $t \in (10, 15)$. During this time period, both teams play in a similar style, and their rate functions almost overlap.

Due to the 24-second time limit for each play, both team increase their offensive pace after $t > 15$. However, when time reaches about $t \approx 20$, the two teams start to show highly distinctive playing patterns. For the Heat, all three of their rate functions rise rapidly, which means that all of their players tend to release the ball quickly, either passing it on to others or shooting. For the Celtics, their $\lambda_2(t)$ and $\lambda_3(t)$ also rise, but not as much as those of the Heat. The Celtics appear to play with more patience. An unusual phenomenon

Table 3.4: Estimated transition probabilities (P_{kl} and P_{ka}) for the 2011-2012 Miami Heat and Boston Celtics ($K = 3$). Rows are originating clusters and columns are receiving clusters and play outcomes.

		C1	C2	C3	Make2	Miss2	Make3	Miss3	Fouled	TO
Heat	C1	0	0.564	0.296	0.035	0.014	0	0.042	0.021	0.028
	C2	0.188	0.262	0.225	0.103	0.087	0.008	0.032	0.063	0.032
	C3	0.226	0.426	0.090	0.052	0.064	0.039	0.064	0.013	0.026
Celtics	C1	0.175	0.332	0.327	0.031	0.083	0.010	0.016	0.005	0.021
	C2	0.270	0.066	0.262	0.065	0.172	0.033	0.066	0.041	0.025
	C3	0.304	0.177	0.094	0.191	0.149	0	0.014	0.057	0.014
Together	C1	0.119	0.479	0.250	0.032	0.053	0.006	0.026	0.012	0.023
	C2	0.220	0.210	0.211	0.097	0.124	0.014	0.039	0.056	0.029
	C3	0.262	0.341	0.083	0.110	0.087	0.023	0.045	0.030	0.019

is that, for the Celtics, their rate function $\lambda_1(t)$ actually decreases after $t > 17$. This is because the starting point guard, Rajon Rondo (in cluster 1), is not the best jump shooter. Close to the end of the time limit and against the tough defense from the Heat, he typically struggles a bit trying to pass or shoot, so the ball stays in his hands for a little longer.

In Appendix A.5, we provide an expanded version of Figure 3.4 which includes 95% confidence bands for these rate functions.

Transition probabilities The estimated transition probabilities for events originating from the three different clusters are presented in Table 3.4. We will focus on the transition probabilities of each team alone. When the two teams are pooled together, the estimated transition probabilities simply appear to be averages of the individual team results.

First, we look at passes among clusters. For the Heat, James and Wade (both in cluster 2) are the absolute key players for the team, so players from both cluster 1 and cluster 3

tend to pass the ball to them (cluster 2) with very high probabilities (56.4% and 42.6%, respectively). James and Wade also pass the ball more often to each other than to the other clusters (26.2% vs. 18.8% and 22.5%, respectively). The two players in cluster 1, Chalmers and Cole, do not pass to each other in our data because they are never on the court at the same time during those games. The Celtics, on the other hand, tend to move the ball more evenly among the three clusters. Their clusters 1 and 2 each has almost equal probabilities to pass the ball to the other two clusters. Their transition probabilities are lower within each cluster than between different clusters.

Next, we discuss shooting choices. For the Heat, the overall probabilities of shooting the ball (sum of Make 2, Miss 2, Make 3, and Miss 3) are 9.1% for cluster 1, 23% for cluster 2, and 21.9% for cluster 3. Meanwhile, the corresponding numbers for the Celtics are 14.0% for cluster 1, 33.6% for cluster 2, and 35.4% for cluster 3. Relatively speaking, when releasing the ball, the Heat players have lower chances to take a shot than the Celtics players do, but higher chances to pass the ball to their teammates. This shows the offense of the Heat involves more interactions among players. For both teams, the respective shooting probabilities for clusters 2 and 3 are more than twice as high as those for cluster 1. Let us look into these probabilities in more detail. James and Wade (cluster 2, Heat) shoot many more 2-pointers than 3-pointers, and incredibly, they score more than half of their 2-pointer shots. Indeed, James and Wade are outstanding at penetration, but not great 3-point shooters. By contrast, Pierce and Allen (cluster 2, Celtics) are better balanced. They shoot and make more 3-pointers than James and Wade do. In the offensive end, Pierce has been regarded as one of the most well-rounded players (as of 2012), because of his ability to score from almost any location. Allen is an extraordinary 3-point shooter —

actually one of the best in the entire NBA history. Unfortunately, Pierce and Allen miss many 2-pointers in these two games. For the Heat, both their cluster 1 and cluster 3 shoot many 3-pointers (almost as many as 2-pointers), since one of their main strategies is for James and Wade to attract the defense from their opponents while their other players seek open-shot opportunities (mostly 3-pointers). For the Celtics, their clusters 1 and 3 mostly shoot 2-pointers, and their main attacking areas are close to the hoop.

Finally, we examine the probabilities of drawing a foul and committing a turnover. Note that "drawing a foul" means being fouled by the opposing team, often after fooling them with fake moves. For the Heat, James and Wade draw fouls with much higher probability than do their teammates in cluster 1 and cluster 3 (6.3% vs. 2.1% and 1.3%). The reason is that James and Wade are often the ones to penetrate, while their teammates usually play "catch and shoot". For the Celtics, players in their cluster 3 have the highest probability of drawing fouls, because those players — for example, Kevin Garnett — are very aggressive when playing close to the hoop; players in their cluster 2 are also good at drawing fouls, as Pierce is a master at doing so. Overall, the Celtics are more capable of drawing fouls, but they make fewer turnovers than the Heat, because they play at a slower pace and make fewer passes.

3.3.3 Cleveland Cavaliers versus Golden State Warriors in 2015

We now analyze two games in the 2015 NBA finals between the Cleveland Cavaliers and the Golden State Warriors — in particular, games 2 and 5. Again, we consider only the first three quarters. These two games are particularly interesting case-study materials for

us because there was a fascinating change in the Warriors' lineup in between. After losing both games 2 and 3 of the series, Steve Kerr, the head coach of the Warriors, decided to change their regular lineup to a small lineup, which meant that they stopped playing centers. This was an unconventional strategy but it successfully turned the series around, and the Warriors went on to win the championship that year by winning three consecutive games!

These two teams have very different styles of play. The aforementioned change in the Warriors' lineup meant there was a big change in how the two teams played these two particular games as well. Thus, unlike in the previous section, in this section we simply fit four CSBMs separately for each team and each game, and no longer fit a pooled model combining the two teams and the two games together. Overall, there are four data sets. For game 2, the Cavaliers have eight players, 84 plays and 290 transactions, while the Warriors have ten players, 75 plays and 307 transactions. For game 5, the Warriors have ten players, 79 plays and 296 transactions, whereas the Cavaliers have eight players, 81 plays and 291 transactions. As in the previous section, in what follows we give detailed discussions about the clustering results, initial probability estimates, fitted rate functions, and transition probability estimates, in that order.

Clustering results The cluster labels of the players for the two games are reported in Table 3.5. As in the previous section, we set $K = 3$ here as well.

For the Cavaliers, the results from the two games are similar, except their two shooting guards — Iman Shumpert and J.R. Smith — switch clusters. It is not surprising that

Table 3.5: Clustering results for the 2014-2015 Cleveland Cavaliers and Golden State Warriors ($K = 3$). Cluster labels are C1, C2, C3. Four different clustering results are presented (two teams $\times$ two games). Player positions are included for reference only; they are not used by the clustering algorithm.

			Game 2			Game 5		
			Alone			Alone		
Team	Player	Position	C1	C2	C3	C1	C2	C3
Cavaliers	Matthew Dellavedova	PG		X			X	
	Iman Shumpert	SG		X				X
	J.R. Smith	SG			X		X	
	LeBron James	SF	X			X		
	James Jones	SF			X			X
	Mike Miller	SF			X			X
	Tristan Thompson	PF			X			X
	Timofey Mozgov	C			X			X
Warriors	Stephen Curry	PG	X			X		
	Shaun Livingston	PG	X			X		
	Klay Thompson	SG		X			X	
	Leandro Barbosa	SG		X			X	
	Harrison Barnes	SF			X		X	
	Andre Iguodala	SF	X					X
	Draymond Green	PF	X					X
	David Lee	PF	Did	Not	Play		X	
	Andrew Bogut	C			X	Did	Not	Play
	Festus Ezeli	C			X	Did	Not	Play
	Marreese Speights	C			X	Did	Not	Play

LeBron James is in a cluster by himself. In these two games, he is the only core player of the Cavaliers since their other two superstars, Kyrie Irving and Kevin Love, are both absent due to injuries. Without support from other superstar teammates, James has to take charge of a large amount of ball handling, passing and scoring; he simply does it all. Indeed, James is one of the most versatile players in the history of the NBA. With James being the only *primary* ball handler of the Cavaliers, their cluster 2 consists of *secondary* ball handlers: the point guard, Matthew Dellavedova, for both games; and a shooting guard — Shumpert for game 2 and Smith for game 5. In general, both Shumpert and Smith can dribble and shoot. Shumpert handles the ball more often than does Smith in game 2, but their roles are reversed in game 5. Other than Smith (in game 2) and Shumpert (in game 5), their cluster 3 consists of {James Jones, Mike Miller}, both catch-and-shoot players, and {Tristan Thompson, Timofey Mozgov}, both inside (the paint) players. Overall, the Cavaliers are a team built around a single key player, LeBron James.

The Warriors, on the other hand, play the two games in fairly different styles. First of all, the active rosters are different: all three centers — Andrew Bogut, Festus Ezeli and Marreese Speights — play in game 2 but not in game 5; meanwhile, David Lee does not play in game 2, but does play in game 5. We already explained the reason behind these changes in their lineup at the beginning of this section (Section 3.3.3). Beyond the clear change of rosters, our CSBM reveals more insight into the different playing styles of the Warriors in these two games. Unlike the Cavaliers, the Warriors have 4 primary ball handlers and distributors: Stephen Curry (PG), Shaun Livingston (PG), Andre Iguodala (SF) and Draymond Green (PF). In game 2 under their regular lineup, our model clusters these four players together. The two shooting guards, Klay Thompson and Leandro Barbosa, are

Table 3.6: Estimated transition probabilities (P_{sk}) from each initial action to clusters C1, C2 and C3, for four different clustering models of the 2014-2015 Cleveland Cavaliers and Golden State Warriors.

		C1	C2	C3
Cavaliers Game 2	Inbound	0.489	0.422	0.089
	Rebound	0.265	0.088	0.647
	Steal	0.200	0.400	0.400
Cavaliers Game 5	Inbound	0.500	0.409	0.091
	Rebound	0.429	0.107	0.464
	Steal	0.286	0.428	0.286
Warriors Game 2	Inbound	0.767	0.093	0.140
	Rebound	0.400	0.160	0.440
	Steal	0.714	0.143	0.143
Warriors Game 5	Inbound	0.660	0.140	0.200
	Rebound	0.185	0.296	0.519
	Steal	0.250	0	0.750

clustered in one cluster. The three centers together with a small forward, Harrison Barnes, form the last cluster. In game 5 under their small lineup, our model divides their 4 primary ball handlers into two clusters — the two point guards, Curry and Livingston, are in one cluster; the two forwards, Iguodala and Green, are in another. All remaining players are in a separate cluster. Note that, although both Barnes and Lee are forwards, their roles in the team are considerably less important than those of Iguodala and Green.

Initial probabilities The estimated transition probabilities from each initial action to the three clusters are shown in Table 3.6.

For the Cavaliers, the probabilities of the two games are similar, except the rebounds of LeBron James (the only player in cluster 1). James catches many more rebounds in game

5 than he does in game 2 (42.9% vs. 26.5%). The reason here is that, with the Warriors playing the small lineup, James becomes one of the tallest and biggest men on the court, playing closer to the rim and catching more rebounds. For both games, more than 90% of the inbounds go to cluster 1 and cluster 2, with cluster 1 receiving slightly more than cluster 2. Players in cluster 2 contribute more than 40% of the steals in the two games, while the other two clusters split the remainder.

For the Warriors, recall that their three centers, belonging to cluster 3 in game 2, do not play in game 5, and their two forwards, Iguodala and Green, belonging to cluster 1 in game 2, become the new cluster 3 in game 5. As a result, their inbound probabilities change slightly, but their rebound and steal probabilities change dramatically. To get into more details, their players in cluster 1 have a much higher probability of receiving an inbound than those in the other two clusters combined, because their cluster 1 contains two point guards, Curry and Livingston. However, this probability goes down by about 10% from game 2 (76.7%) to game 5 (66%), whereas those of cluster 2 and cluster 3 each increases about 5%. These results imply that, when the Warriors switch to their small lineup in game 5, players other than those in cluster 1 also get more opportunities to receive inbounds and initiate plays. In game 5, due to the absence of centers, who make up cluster 3 and contribute 44% of the rebounds in game 2, all players start to share their contributions to catching rebounds as well. In particular, Green and Iguodala (in cluster 3 for game 5) now catch 51.9% of the rebounds, in contrast to $< 40\%$ when they are in cluster 1 for game 2; the contribution of cluster 2 to rebounds increases from 16% in game 2 to 29.6% in game 5; and finally, without Green and Iguodala (now in cluster 3), the two point guards that remain in cluster 1 (i.e., Curry and Livingston) also manage to catch 18.5% of the

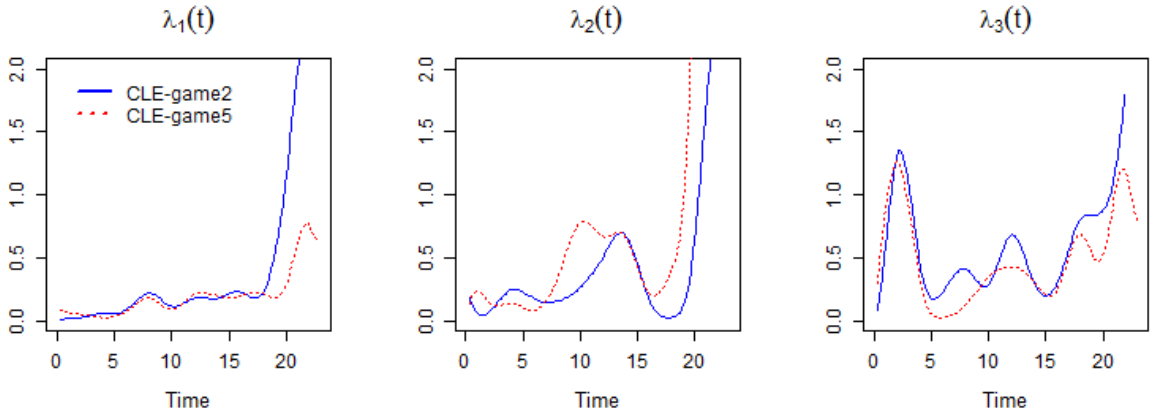


Figure 3.5: Fitted rate functions for the 2014-2015 Cavaliers, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rates with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively.

rebounds. Regarding steals, the most significant changes are a huge decrease for cluster 1 (71.4% to 25%) and a huge boost for cluster 3 (14.3% to 75%). Once more, this is because Green and Iguodala have "moved" from cluster 1 to cluster 3; they both are top defenders who contribute to many steals.

Rate functions The fitted rate functions of the Cavaliers and the Warriors are displayed in Figure 3.5 and Figure 3.6, respectively.

For the Cavaliers, the rate functions from the two games appear to be generally similar for each respective cluster, with some small differences. For cluster 1 (James), its rate function $\lambda_1(t)$ is almost the same in the two games for $t < 17$ — fairly flat and low. This means that James plays with almost the same style at the beginning of a play in both games, keeping the ball in his hands and organizing the offense. Toward the end of a play, James

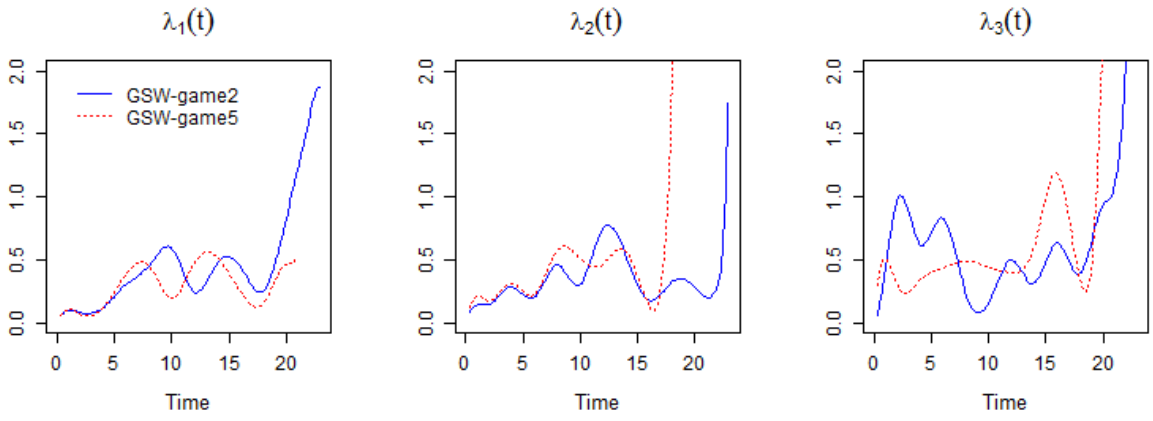


Figure 3.6: Fitted rate functions for the 2014-2015 Golden State Warriors, $\lambda_1(t)$, $\lambda_2(t)$ and $\lambda_3(t)$, each describing the rates with which the ball leaves a player in cluster 1, cluster 2 and cluster 3, respectively.

starts to "heat up" at around $t \approx 17$ in game 2, whereas he does so slightly later in game 5, at about $t \approx 19$. This is because the small lineup of the Warriors in game 5 move much more quickly, so they can defend James more effectively in the last few seconds and delay his offense. For cluster 2, the first big difference appears after $t > 7$. In game 2, $\lambda_2(t)$ grows *slowly* to reach a peak at $t \approx 14$; however, in game 5, the same function $\lambda_2(t)$ grows *rapidly* after $t > 7$ and maintains a high level until $t \approx 14$. Clearly, players in cluster 2 have increased their offensive pace in game 5. On the one hand, Smith (cluster 2 SG in game 5) does more quick-release shooting than does Shumpert (cluster 2 SG in game 2). On the other hand, the higher defensive pressure created by the Warriors' small lineup has forced the Cavaliers to move the ball more quickly. For the same reasons, toward the end of a play, players in cluster 2 also tend to attack the rim or pass the ball slightly earlier in game 5 (at $t \approx 16$) than they do in game 2 (at $t \approx 18$). For cluster 3, their rate function $\lambda_3(t)$ displays a similar pattern in the two games, but the one in game 5 is

almost entirely dominated by the one in game 2. Players in this cluster are big men and typically catch-and-shoot players; they are usually not responsible for handling the ball. They catch rebounds and start a play by passing the ball to their teammates in the other two clusters. At around $t \approx 12$, they get their first chance to touch the ball, when they either shoot or pass it back to the ball handlers. Their second chance to touch the ball happens near the end of a play, when they have to shoot rapidly. In game 5, the small lineup of the Warriors can quickly cover the open shots and "double up" to defend a big man in the paint, and that forces players in Cavaliers' cluster 3 to keep the ball in their hands for a slightly longer period. This is why their $\lambda_3(t)$ is lower in game 5 than in game 2. Overall, the patterns displayed in the Cavaliers' three rate functions are quite similar in the two games. The changes mostly can be attributed to the different defensive strategies used by their opponent.

For the Warriors, though, due to the change in their lineup, their rate functions from the two games are noticeably different. In game 2 with their regular lineup, their rate functions (blue solid lines in Figure 3.6) show regular patterns — at the start of a play, $\lambda_1(t)$ and $\lambda_2(t)$ are relatively low, while $\lambda_3(t)$ has high peaks. This means that, at the start of a play, players in cluster 1 and cluster 2 tend to handle the ball, whereas those in cluster 3 catch rebounds and pass the ball out more or less immediately. This is the same as the playing style of the Cavaliers. However, their rate functions have more peaks than those of the Cavaliers. Moreover, their $\lambda_1(t)$ and $\lambda_2(t)$ in game 2 are, in general, higher than those of the Cavaliers at the start of a play. These show that the Warriors' offense is more flexible — the ball is passed more frequently, so everybody gets chances to touch it, and no one holds the ball for a very long time. In fact, this has become the Warriors'

signature team-playing style. However, in game 5, all three of their rate functions show significant differences. First, the two peaks of $\lambda_1(t)$ occur earlier in game 5 than in game 2. Second, the rapid growth of $\lambda_2(t)$ also appears earlier in game 5 (at $t \approx 17$) than in game 2 (at $t \approx 22$). Both differences indicate that, with a small lineup, the Warriors have increased their offensive pace in game 5. Finally, their $\lambda_3(t)$ changes dramatically between the two games; in game 5, it is much flatter at the beginning and has a much higher peak at $t \approx 17$. This is certainly because, in game 5, the players making up cluster 3 are entirely different from the ones in game 2. From Table 3.6, we know that their cluster 3 in game 5 (Green and Iguodala) catch a larger proportion of rebounds than do their cluster 3 in game 2 (three centers), but instead of immediately passing the ball out, Green and Iguodala both often dribble and run the play. The peak of $\lambda_3(t)$ at $t \approx 17$ in game 5 is particularly significant, revealing one key offensive strategy of the Warriors' small lineup, the so-called "high pick-and-roll". A typical sequence of this strategy is as follows: Curry dribbles the ball outside the three-point line, and Green (or Iguodala) comes to set up a screen (a "human body wall"). Thanks to Curry's incredible three-point shooting skills, after he dribbles around the screen both defenders of Curry and of Green (or Iguodala) usually have to focus on covering Curry together, leaving Green (or Iguodala) wide open, so Curry can now pass the ball to him. Green (or Iguodala) can then shoot the ball; drive to the basket directly; or take one or two dribbles, draw another defender, and then pass the ball to another wide-open teammate, who is usually waiting at the three-point line on the weakly-defended side. This entire sequence often happens very quickly within three seconds.

Overall, the estimated rate functions reveal many intricate details of a team's playing style.

Table 3.7: Estimated transition probabilities (P_{kl} and P_{ka}) for the 2014-2015 Cleveland Cavaliers and Golden State Warriors ($K = 3$). Rows are originating clusters and columns are receiving clusters and play outcomes.

		C1	C2	C3	Make2	Miss2	Make3	Miss3	Fouled	TO
Cavaliers Game 2	C1	0	0.167	0.430	0.111	0.153	0.014	0.014	0.069	0.042
	C2	0.292	0.141	0.259	0.016	0.081	0	0.114	0.032	0.065
	C3	0.335	0.160	0.111	0.098	0.123	0.037	0.037	0.062	0.037
Cavaliers Game 5	C1	0	0.384	0.274	0.123	0.110	0	0.027	0.027	0.055
	C2	0.296	0.181	0.261	0.024	0.036	0.059	0.107	0	0.036
	C3	0.346	0.198	0.076	0.061	0.122	0.045	0.061	0.061	0.030
Warriors Game 2	C1	0.357	0.233	0.194	0.037	0.037	0.007	0.060	0.030	0.045
	C2	0.380	0	0.120	0.140	0.080	0.100	0.120	0.060	0
	C3	0.469	0.226	0	0.061	0.081	0	0.061	0.061	0.041
Warriors Game 5	C1	0.159	0.276	0.323	0.058	0.058	0.046	0.034	0	0.046
	C2	0.151	0.097	0.285	0.151	0.166	0.015	0.015	0.060	0.060
	C3	0.330	0.267	0.137	0.064	0.038	0.025	0.038	0.076	0.025

The Cavaliers play around their key superstar, LeBron James, whereas the Warriors share the ball more evenly. It also can be easily seen that the Warriors have played these two games quite differently and the Cavaliers have responded with small but clear adjustments in their playing style as well.

Transition probabilities The estimated transition probabilities of events originating from the three different clusters are displayed in Table 3.7, for both the Cavaliers and the Warriors.

For the Cavaliers, the overall probabilities to pass the ball (sum of the first three columns) for the three respective clusters are {59.7%, 69.2%, 60.6%} in game 2, and {65.8%, 73.8%, 62%} in game 5. Clearly, the Cavaliers make more passes in game 5 than in game 2,

which is due to the stronger defense by the Warriors' small lineup. For the same reason, in game 2 James (the only player in cluster 1) passes more to cluster 3 (shooters and big men), whereas in game 5 he passes more to cluster 2 (ball handlers). The respective roles of their cluster 2 and cluster 3 do not change much in the two games — cluster 2 is the bridge between cluster 1 and cluster 3, making almost an equal proportion of passes to each of the other two clusters; cluster 3, however, more often passes the ball back to James (cluster 1). The overall probabilities to shoot the ball (sum of columns 4-7) for the three respective clusters are {29.2%, 21.1%, 29.5%} in game 2, and {26%, 22.6%, 28.9%} in game 5, which do not change much. When facing the quick defense of the Warriors in game 5, the Cavaliers have successfully created an almost equal percentage of shots by making more passes. Regarding the probabilities of being fouled and making turnovers, James (cluster 1) fails to draw as many fouls in game 5 as he does in game 2 (2.7% vs. 6.9%), but he makes more turnovers (5.5% vs. 4.2%). These can be partly attributed, again, to the stronger defense by the Warriors' small lineup, especially the one-on-one defense on James by Iguodala. Players in cluster 2 are not as aggressive in game 5 as they are in game 2 — although they make fewer turnovers (3.6% vs. 6.5%), they do not draw any fouls at all (0% vs. 3.2%). The performance of cluster 3 is fairly stable in the two games in terms of drawing fouls and making turnovers.

For the Warriors, the overall passing probabilities of their three respective clusters are {78.4%, 50%, 79.5%} in game 2, and {75.8%, 53.3%, 73.4%} in game 5. Despite the drastic changes in their lineup, these probabilities do not change much. Each of the first three columns in Table 3.7 contains the probabilities that the corresponding cluster is the receiver of the ball passed from different clusters. Here, we can easily see that a considerable

proportion of the passes have shifted from cluster 1 to cluster 3 in game 5. This is because Green and Iguodala, two of the four primary ball handlers, are now in cluster 3 as opposed to cluster 1, and they receive many passes. The overall shooting probabilities (sum of columns 4-7) for their three respective clusters are {14.1%, 44%, 20.3%} in game 2, and {19.6%, 34.7%, 13.5%} in game 5. In both games, players in cluster 2 are more likely to shoot than those in the other two clusters. This makes sense because cluster 2 contains two shooting guards, Klay Thompson and Leandro Barbosa, who both are excellent scorers and often take on a huge responsibility in shooting the ball. It can also be seen that, in game 5, the probability to shoot has increased for cluster 1 but decreased for cluster 2. This is because the small lineup gives players in cluster 1 — especially Curry — more open space and hence better shooting opportunities; by contrast, Klay Thompson (cluster 2), who is less affected by the change in the lineup, struggles with shooting in game 5. For cluster 3, we see that Green and Iguodala (cluster 3 in game 5) are less likely to shoot than the centers (cluster 3 in game 2). With regard to shooting, it is well-known that the Warriors rely on three-pointers as one of their most important scoring methods. Curry and Thompson are arguably the best three-point shooting back-court duo in the entire history of the NBA. From Table 3.7, we can clearly see that the Warriors attempt many more three-pointers than the Cavaliers do and they also succeed more often. One surprising observation is that players in their cluster 2 shoot considerably fewer three-pointers in game 5 than they do in game 2. Indeed, this is another piece of evidence showing the struggle of Klay Thompson in game 5. There are two significant differences in terms of drawing fouls and making turnovers: cluster 1 fails to draw any fouls in game 5 versus 3% in game 2; and cluster 2 makes more turnovers in game 5 than in game 2 (6% vs. 0%).

3.3.4 LeBron James: Miami Heat versus Cleveland Cavaliers

Both the 2011-12 Miami Heat and the 2014-15 Cleveland Cavaliers had LeBron James on their teams and made him the key player. Thus, it is especially interesting for us to compare the player structures of these two teams, and to see if there is any difference in how James has played the game with these different teams. We investigate the first question by pooling the transactions of the Heat (in their two 2012 games versus the Celtics) and the transactions of the Cavaliers (in their two 2015 games versus the Warriors), and applying the CSBM to cluster the players from both teams together. With regard to the second question, we simply compare the individual results we have obtained earlier for the Heat (Section 3.3.2) and for the Cavaliers (Section 3.3.3).

For the pooled CSBM, we focus primarily on the clustering results in this section and forsake any detailed discussions of the rate functions or the transition probabilities. Other than LeBron James, Mike Miller and James Jones are also on both of these teams. When playing on different teams, the same player may play in a different style, depending on his specific role for the team. Hence for James (and likewise for Miller and Jones, too), we create two separate avatars — one for the games he played on the Heat and another for the games he played on the Cavaliers — and treat them as two different "players" in the clustering algorithm. We are especially curious whether the pooled CSBM will cluster the two avatars of the same player into the same cluster or different clusters.

Table 3.8 displays the clustering results from the pooled CSBM, fitted to all transactions of the Heat and the Cavaliers in the 4 games we have annotated. With a total of 19 "players", we now choose $K = 4$ instead of $K = 3$ as we did in the previous two sections; this allows

Table 3.8: Clustering results for the 2011-2012 Miami Heat and the 2014-2015 Cleveland Cavaliers together ($K = 4$). Cluster labels are C1, C2, C3, C4. Players appearing with two separate avatars for the clustering algorithm are bolded. Player positions are included for reference only; they are not used by the clustering algorithm.

			Together			
Team	Player	Position	C1	C2	C3	C4
Heat	Mario Chalmers	PG	X			
	Norris Cole	PG	X			
	Dwyane Wade	SG		X		
	LeBron James	**SF**		**X**		
	James Jones	**SG**			**X**	
	Shane Battier	SF			X	
	Mike Miller	**SF**			**X**	
	Chris Bosh	PF				X
	Udonis Haslem	PF				X
	Ronny Turiaf	C				X
	Joel Anthony	C				X
Cavaliers	Matthew Dellavedova	PG	X			
	Iman Shumpert	SG			X	
	J.R. Smith	SG			X	
	LeBron James	**SF**		**X**		
	James Jones	**SF**			**X**	
	Mike Miller	**SF**			**X**	
	Tristan Thompson	PF				X
	Timofey Mozgov	C				X

us to cluster the "players" with a slightly finer resolution.

Our clustering results clearly indicate that the 2011-12 Heat and the 2014-15 Cavaliers are built in a very similar way. Cluster 1 consists of point guards; cluster 2 consists of superstars — namely, LeBron James (for both teams) and Dwyane Wade (for the Heat); cluster 3 consists of the other perimeter players — mostly shooters and perimeter defenders; and the last cluster is made up of big men — power forwards and centers. It also turns out that the two avatars of the same player (whether James, Miller or Jones) are always clustered together. Indeed, both teams are built around LeBron James and their playing styles are similar, too. James is the primary ball handler and distributor for both teams. While playing for the Heat, James has Wade as an important helper, but while playing for the Cavaliers, he is the only superstar. We can imagine that, if Kyrie Irving, the superstar point guard of the Cavaliers, were not injured, he might have joined James and Wade in cluster 2. The point guards in these two team are secondary ball handlers and serve as bridges between the superstars and the other players. Players in cluster 3 are mainly responsible for playing defense and "catch and shoot". The big men in cluster 4 are mostly responsible for catching rebounds and scoring under the rim.

In the rest of this section, we revisit some individual results for the Heat (Section 3.3.2) as well as for the Cavaliers (Section 3.3.3) in order to compare in more detail the performance of LeBron James in those two series.

First, recall that our cluster labels (e.g., C1, C2, ...) are arbitrary, and that James has been clustered into C2 with the 2011-12 Heat but into C1 with the 2014-15 Cavaliers. Comparing Figure 3.4 (middle panel) and Figure 3.5 (left panel), we find that the Heat's

$\lambda_2(t)$ function has more peaks and is higher than the Cavaliers' $\lambda_1(t)$ function overall. This shows that, while playing for the Heat, James chooses to pass the ball more often at the beginning of a play. This is mostly because of the presence of Wade, a superstar teammate, who interacts with James more frequently than the point guards. Actually, the "two-man fast break" by James and Wade is one of the Heat's defining features. Second, comparing Table 3.3 and Table 3.6, we find that, while playing for the Heat, James and Wade *together* receive 19.4% of the inbounds, whereas, while playing for the Cavaliers, James *alone* receives a staggering 50% of the inbounds. The Heat mostly let their point guards carry the ball past the half court, because they always have one of them (either Chalmers or Cole) on the court. However, with Irving out on injury, the Cavaliers only play one point guard (Dellavedova) in their lineup, so James has to carry the ball more than usual. Third, while on the Heat, James and Wade *together* average a 23% probability to shoot, but while on the Cavaliers, James *alone* has an even higher probability to shoot — 29.2% and 26% respectively in the two games against the Warriors. James is a great scorer as well as offensive organizer. He can freely switch between these two modes of play depending on the situations in the game. While playing for the 2011-12 Heat, James has stronger teammates, so he tends to create more shooting opportunities for others. With the 2014-15 Cavaliers, however, James must take more shots by himself due to the limited support from his teammates.

In summary, our analysis using the CSBM shows that the player structure of the 2011-12 Heat and that of the 2014-15 Cavaliers are fairly similar. The CSBM also reveals many subtle differences in LeBron James' playing style in the two series.

3.4 Summary and Remarks

In this chapter, we advocate the concept that basketball games can be analyzed as transactional networks. We have proposed a multistate Continuous-time Stochastic Block Model to cluster players based on their styles of handling the ball. In particular, we model each basketball play as an inhomogeneous continuous-time Markov chain, with the transition rate functions being governed by the players' cluster memberships. We adopt B-splines to model the rate functions and develop an EM^+ algorithm to estimate model parameters. Applications to a number of NBA games between the 2011-12 Miami Heat and Boston Celtics and between the 2014-15 Cleveland Cavaliers and Golden State Warriors have yielded compelling evidence that the CSBM framework is of great practical value in clustering and evaluating basketball players.

As the popularity of basketball analytics appears to be growing in recent years, it is perhaps helpful for us to summarize the main differences between our work and a few recent works in this area (e.g., Fewell et al., 2012; Cervone et al., 2016). The key features of our work are: (i) viewing basketball games from a network perspective, (ii) consideration of time dynamics, and (iii) clustering of players at an individual level. In what follows, we discuss how our work differs from a few others in terms of these features; a brief summary is given in Table 3.9.

Fewell et al. (2012) certainly view basketball games from a network perspective as well, but they do not take time dynamics into account, and their treatment of players occurs at a position level rather than an individual level. Specifically, they pre-group players according

Table 3.9: Summary of differences between our work and others.

	Network Perspective	Time Dynamics	Model Objective
CSBM	yes	yes	descriptive at individual level
Fewell et al. (2012)	yes	no	descriptive at position level
Cervone et al. (2016)	no	yes	predictive at individual level (of final point outcome)

to their on-court positions (e.g., point guard, shooting guard, and so on). Whereas our CSBM describes player differences based on the real-time dynamics of how each basketball play unfolds, the method developed by Fewell et al. (2012) aims to describe differences in how each of the five pre-defined positions communicates with each other — and with various initial and absorbing states — at an aggregate level, aggregated over both all players holding the same position and all transactions during a certain time period (e.g., an entire game). In their work, point guards are always considered together with other point guards, and any player difference at the individual level is suppressed. While it is hardly surprising that many players holding the same position often end up being clustered together by our CSBM, this is certainly not always the case. For example, our analysis of the two games between the 2014-15 Cleveland Cavaliers and Golden State Warriors (Section 3.3.3) clearly shows that the distinctive playing style of LeBron James almost calls for the definition/creation of a new on-court position, for which some long-time basketball observers have informally suggested the name of "point forward". Our analysis also shows that players like Draymond Green (a power forward) and Andre Iguodala (a small forward) are certainly playing important roles in the game beyond the traditional ones defined by their respective on-court positions.

Cervone et al. (2016), on the other hand, do consider time dynamics, but they do not view basketball games from a network perspective. While they track the movement of the ball both spatially and over time, they do not view players as nodes and passes as edges. Most importantly, their objective is fundamentally different from ours. Our goal is to cluster players according to their individual playing styles as characterized by the rate functions $\lambda_k(t)$ and the transition probabilities P_{sk}, P_{kl}, and P_{ka}, but theirs is to predict the final point value of each basketball play/possession as the individual play unfolds. One can say that their analysis is driven by *outcome* but ours is driven by *style*. Although rate functions for ball passing are components of both models, their structure and role vary considerably. Our rate functions are smooth functions of clock time, and are used to characterize groups of players with similar transition rates. Their rate functions are log-linear regressions which use predictors derived from motion-capture data, forming one component in a hierarchical model whose ultimate objective is to predict point value. They are not used to cluster players.

Chapter 4

Variable Selection Networks

4.1 Introduction

Recent developments of technology have led to an explosion of high dimensional data in many disciplines, including health science, finance, economics, engineering, etc. The number of variables can be thousands, tens of thousands or even millions. Nevertheless, only very few variables are believed to have true influence on the response. Hence, variable selection becomes a fundamentally important problem. Due to the high dimensionality, the traditional best subset selection becomes computationally infeasible. A large amount of research works have been devoted to the high dimensional variable selection over the past two decades.

In the field of statistics, penalized likelihood methods have gained enormous attention, for example, the nonnegative garrotte (Breiman, 1995), the LASSO (Tibshirani, 1996), the

SCAD (Fan and Li, 2001), the elastic net (Zou and Hastie, 2005), the Adaptive LASSO (Zou, 2006) and the Dantzig selector (Candes and Tao, 2007). Their theoretical properties, algorithms and generalizations have been extensively studied. Fan and Lv (2010) and Buhlmann and van de Geer (2011) provide comprehensive overviews of the penalized likelihood methods for high dimensional variable selection. During the last few years, many screening methods have been proposed to handle the more challenging situation of the ultra-high dimensionality (Fan and Lv, 2008; Wang, 2009; Jin et al., 2014; Cho and Fryzlewicz, 2012; Ma et al., 2016). They first employ a variable screening procedure to hugely reduce the number of variables and then apply a variable cleaning procedure to conduct variable selection. At the same time, Bayesian variable selection has also become a very active field. Many recently developed approaches have shown convincing performance, for example, Narisetty and He (2014), Bondell and Reich (2012) and Johnson and Rossell (2012). A review of Bayesian Variable Selection can be found in O'Hara and Sillanpaa (2010).

Inspired by the concept of ensemble learning, Xin and Zhu (2012) propose Variable Selection Ensemble (VSE), a novel framework for variable selection. Ensemble learning techniques, for instance, boosting, bagging, Random Forest, etc, have been widely adopted for estimation and classification. They are practically easy to implement and show outstanding performance. Now for variable selection, given p variables, there are overall 2^p submodels. The main idea of the VSE is to evaluate a number of submodels and incorporate the overall information to select a final model, where the final model is not necessarily among those checked submodels. The group of submodels to be evaluated can be chosen stochastically, for example, Xin and Zhu (2012) and Zhu and Chipman (2006), or systematically, as what

we will do in this work, where we consider all $p(p-1)/2$ submodels containing all pairs of variables. If each variable is treated as a node, such a VSE is actually a network, namely, a *Variable Selection Network*.

We first introduce variable selection networks (VSN) and algorithms for the $p < n$ case in Section 4.2 to 4.4. In Section 4.2, a VSN with binary (0/1) edges is constructed and its theoretical properties are established. In Section 4.3, a weighted VSN is briefly discussed. Three VSN algorithms are proposed in Section 4.4. To handle the $p \geq n$ scenario, in Section 4.5, we propose an iterative group screening algorithm. In Section 4.6 and 4.7, the performance of VSN methods is investigated through simulation studies and illustrated by an real data example, respectively. Finally, we summarize our work in Section 4.8.

4.2 A Binary Variable Selection Network

We consider the linear regression:

$$Y = X\beta + \epsilon, \tag{4.1}$$

where X is an $n \times p$ design matrix, $\beta = (\beta_1, \beta_2, \ldots, \beta_p)^T$ are coefficients and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$ are i.i.d. random errors following $N(0, \sigma^2)$.

Given such a linear regression with p covariates, a variable selection network (VSN), represented by a $p \times p$ adjacency matrix $\boldsymbol{A}$, is defined as a network such that the i–j edge A_{ij} is a measure of the importance of variables i and j together. In general, the measure can be any type. We first consider the case that the measure is binary (0/1) valued, where 1

indicates that at least one of the pair of variables is important and 0 means neither of the variables are important. In Section 4.3, we discuss a weighted VSN with edges taking an arbitrary measure.

For the binary VSN, one natural way to evaluate whether variables i and j are important is to conduct the F-test for the following hypothesis (given $p < n$):

$$\mathbf{H}_0 : \beta_i = \beta_j = 0 \text{ vs. } \mathbf{H}_1 : \beta_i \neq 0 \text{ or } \beta_j \neq 0, \tag{4.2}$$

where the true model is assumed to be nested within the full model with p variables. We set $A_{ij} = 1$ if $\mathbf{H}_0$ is rejected; otherwise, $A_{ij} = 0$. In addition, $A_{ii} = 0$ for all $i = 1, 2, \ldots, p$. More explicitly, the test statistic is

$$TS_{ij} = \frac{(RSS_H - RSS)/2}{RSS/(n-p)}, \tag{4.3}$$

where RSS is the residual sum of squares of the full model, and RSS_H is the residual sum of squares of the restricted model under the null hypothesis. Choose a significance level α and let C_α be the $1-\alpha$ quantile of $F(2, n-p)$, the F distribution with degrees of freedom 2 and $n-p$. If $TS_{ij} > C_\alpha$, we reject the null hypothesis and set $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

To further explore the test statistic, we introduce some notation. For a set $\Omega \subseteq \{1, 2, 3, \ldots, p\}$, let X_Ω denote the matrix with columns $\{X_{\cdot\omega} : \omega \in \Omega\}$ and define $X_{-\Omega} = X_{\{1,2,\ldots,p\}/\Omega}$. Given a matrix $\tilde{X}$, define $\mathcal{S}(\tilde{X})$ to be the linear space spanned by the columns of $\tilde{X}$, which is a subspace of the n-dimensional Euclidean space, and define $\mathcal{S}^\perp(\tilde{X})$ to be the orthogonal comple-

ment of $\mathcal{S}(\tilde{X})$. In addition, define $P(\tilde{X}) = \tilde{X}(\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T$, which is the projection matrix to the space $\mathcal{S}(\tilde{X})$. Finally, define $M(\tilde{X}) = I - P(\tilde{X})$, which is the projection matrix to the space $\mathcal{S}^{\perp}(\tilde{X})$. For simplicity, with a slight abuse of notation, we use M_Ω as an abbreviation for $M(X_{-\Omega})$. In particular, $M_{ij} = M(X_{-ij}) = I - P(X_{-ij}) = I - X_{-ij}(X_{-ij}^T X_{-ij})^{-1}X_{-ij}^T$, where $X_{-ij} = X_{\{1,2,\ldots,p\}/\{i,j\}}$.

With the above notation, the test statistic (4.3) can be re-written as

$$TS_{ij} = \frac{Y^T M_{ij} X_{ij}(X_{ij}^T M_{ij} X_{ij})^{-1} X_{ij}^T M_{ij} Y/2}{Y^T(I_n - P(X))Y/(n-p)}. \tag{4.4}$$

Suppose the true model is covered by the full model, we know that

$$\frac{Y^T(I_n - P(X))Y}{\sigma^2} \sim \chi^2(n-p). \tag{4.5}$$

To figure out the distribution of the numerator of TS_{ij} (4.4) (divided by σ^2), note that if $V \sim N(\mu, \Sigma)$ is an m-dimensional multivariate normal random variable and Σ is invertible, then $V^T\Sigma^{-1}V \sim \chi^2(m, \mu^T\Sigma^{-1}\mu)$. Since $X_{ij}^T M_{ij} Y$ is 2-dimensional multivariate normal with $\mathbf{E}(X_{ij}^T M_{ij} Y) = X_{ij}^T M_{ij} X_{ij}(\beta_i, \beta_j)^T$ and $\text{var}(X_{ij}^T M_{ij} Y) = \sigma^2 X_{ij}^T M_{ij} X_{ij}$, suppose $X_{ij}^T M_{ij} X_{ij}$ is invertible, then

$$\frac{Y^T M_{ij} X_{ij}(X_{ij}^T M_{ij} X_{ij})^{-1} X_{ij}^T M_{ij} Y}{\sigma^2} \sim \chi^2(2, \Lambda_{ij}), \tag{4.6}$$

where $\Lambda_{ij} = \frac{1}{\sigma^2}(\beta_i, \beta_j) X_{ij}^T M_{ij} X_{ij}(\beta_i, \beta_j)^T$ is the *noncentrality parameter* of noncentral chi-squared distribution.

Moreover, the numerator and denominator of TS_{ij} are independent because of the fact that $X_{ij}^T M_{ij}(I_n - P(X)) = 0$, which can be easily verified.

Hence,

$$TS_{ij} \sim \frac{\chi^2(2, \Lambda_{ij})/2}{\chi^2(n-p)/(n-p)}, \tag{4.7}$$

with the numerator and denominator being independent. When the null hypothesis is true, i.e., $\beta_i = \beta_j = 0$, we have $\Lambda_{ij} = 0$, so

$$TS_{ij} \sim \frac{\chi^2(2)/2}{\chi^2(n-p)/(n-p)} \sim F(2, n-p); \tag{4.8}$$

when the null hypothesis is false, i.e., $\beta_i \neq 0$ or $\beta_j \neq 0$,

$$TS_{ij} \sim \frac{\chi^2(2, \Lambda_{ij})/2}{\chi^2(n-p)/(n-p)} \sim F(2, n-p, \Lambda_{ij}), \tag{4.9}$$

where $F(2, n-p, \Lambda_{ij})$ is the noncentral F distribution with degrees of freedom 2 and $n-p$, and noncentrality parameter Λ_{ij}.

The probabilities of edges are as follows:

$$\begin{aligned} \mathbf{P}(A_{ij} = 1 | \beta_i = \beta_j = 0) &= \mathbf{P}(TS_{ij} > C_\alpha | \beta_i = \beta_j = 0) \\ &= \mathbf{P}(F(2, n-p) > C_\alpha) \\ &= \alpha \end{aligned} \tag{4.10}$$

and

$$\begin{aligned}\mathbf{P}(A_{ij} = 1 | \beta_i \neq 0 \text{ or } \beta_j \neq 0) &= \mathbf{P}(TS_{ij} > C_\alpha | \beta_i \neq 0 \text{ or } \beta_j \neq 0) \\ &= \mathbf{P}(F(2, n-p, \Lambda_{ij}) > C_\alpha) \\ &:= \alpha_{ij}. \end{aligned} \tag{4.11}$$

All pairs of irrelevant variables have the same probability, α, to form an edge. If at least one variable is true, the probability to form an edge depends on the noncentrality parameter Λ_{ij}. We always have $\alpha_{ij} \geq \alpha$, because, given any $\Lambda_1 \geq \Lambda_2 \geq 0$,

$$\mathbf{P}(\chi^2(r, \Lambda_1) > C) \geq \mathbf{P}(\chi^2(r, \Lambda_2) > C), \tag{4.12}$$

for any degree of freedom r and any constant $C \geq 0$.

4.2.1 Degree Distributions of the Binary VSN

Note that the VSN constructed by the F-test (4.2) is symmetric. Let $d(i)$ denote the degree of variable i such that

$$d(i) = \sum_j A_{ij}. \tag{4.13}$$

In this section, we investigate asymptotic properties of the node degrees of the binary VSN, where 'asymptotic' refers to the number of samples n going to infinity. The number of variables p can also, but not necessarily, go to infinity.

The degrees directly depend on the F-tests that build the VSN. Intuitively, on the one

hand, we want the powers of the tests, i.e., α_{ij}, to approach one asymptotically so that we can effectively detect relevant variables. Clearly, the power of an individual F-test goes to one as long as the noncentrality parameter goes to infinity. However, for the VSN, we are dealing with multiple hypothesis tests, and the number of variables p can go to infinity. Hence, in order to control the overall power, we require the noncentrality parameters $\{\Lambda_{ij} : \beta_i \neq 0 \text{ or } \beta_j \neq 0\}$ to go to infinity at a certain rate. On the other hand, we have to set the significance level $\alpha = \alpha_n$ to approach zero at a certain rate, so that we can avoid selecting irrelevant variables. Overall, in ideal situations, as $n \to \infty$, the edge probabilities of the binary VSN have the following block structure: the relevant variables form a block and the irrelevant ones form another block, and the edge probabilities are approximately

$$\begin{array}{lcc} & \text{relevant} & \text{irrelevant} \\ \text{relevant} & \mathbf{1} & \mathbf{1} \\ \text{irrelevant} & \mathbf{1} & \mathbf{0} \end{array}, \tag{4.14}$$

where a relevant variable has a probability tending to one to have an edge with any other variable and an irrelevant variable has a probability tending to zero to have an edge with any other irrelevant variable.

Define $D = \{j : \beta_j \neq 0\}$ as the set of relevant variables and $s = |D|$ as the number of relevant variables. From the matrix (4.14), we can imagine that the degree of a relevant variable should asymptotically be close to p, whereas the degree of an irrelevant variable should asymptotically be close to s. Due to the common sparsity assumption that $s \ll p$, we can successfully detect relevant variables by their degrees.

In the rest of this section, we will rigorously show that if the noncentrality parameters approach infinity at an order $\lim_{n\to\infty} \Lambda_{ij}/\log n = \infty$, and the significance level is set as $\lim_{n\to\infty} \alpha_n n^2 = \infty$ and $\alpha_n = o(1/p)$, then with a probability tending to one, we can successfully separate the relevant variables from the irrelevant ones by cutting the variable degrees of the VSN from the biggest gap.

Recall that

$$\Lambda_{ij} = \frac{1}{\sigma^2}(\beta_i, \beta_j) X_{ij}^T M_{ij} X_{ij} (\beta_i, \beta_j)^T. \tag{4.15}$$

To assure the asymptotic order of $\{\Lambda_{ij} : \beta_i \neq 0 \text{ or } \beta_j \neq 0\}$, we impose a set of necessary conditions:

(A1) $\psi = \min\{|\beta_i| : \beta_i \neq 0\} > 0$;

(A2) $\lim_{n\to\infty} \min_{i:\beta_i\neq 0} X_i^T M_i X_i/\log n = \infty$, where $M_i = I_n - X_{-i}(X_{-i}^T X_{-i})^{-1} X_{-i}^T$ is the projection matrix to $\mathcal{S}^{\perp}(X_{-i})$.

The first condition sets a lower bound for nonzero coefficients. This is a typical assumption in variable selection literature.

The second condition essentially puts a constraint on the collinearity between any relevant variable i and the other variables. In fact, $M_i X_i$ is the residual vector obtained by regressing a relevant variable X_i over all the other variables (i.e., X_{-i}), and $X_i^T M_i X_i$ is simply the residual sum of squares. The assumption requires the residual sum of squares to grow faster than $\log n$. In other words, the collinearity of X_i and the other columns is not very strong. Chen and Chen (2008) adopt a very similar assumption, which is of the same order. Note that in variable selection literature, a common way to control correlations

among variables is to bound the eigenvalues of X^TX/n, for example, Zhang and Huang (2008) and Wasserman and Roeder (2009). From the fact

$$\frac{X_i^T M_i X_i}{\log n} \geq \phi\Big(\frac{X^TX}{\log n}\Big), \tag{4.16}$$

we can see that assumptions (A2) is weaker than bounding the smallest eigenvalue of X^TX/n.

The following lemma establishes the asymptotic order of Λ_{ij} given the assumptions.

Lemma 1 *If assumptions (A1) and (A2) hold, for $\beta_i \neq 0$ or $\beta_j \neq 0$, we have*

$$\lim_{n\to\infty} \frac{\Lambda_{ij}}{\log n} = \infty. \tag{4.17}$$

Furthermore, the next lemma reveals a connection of the asymptotic order of the noncentrality parameter, the significance level and the power of the test.

Lemma 2 *For $\beta_i \neq 0$ or $\beta_j \neq 0$, suppose $\lim_{n\to\infty} \Lambda_{ij}/\log n = \infty$, and $p < \gamma n$ for a constant $\gamma \in (0,1)$. Let $\lim_{n\to\infty} \alpha_n n^2 = \infty$, then $1 - \alpha_{ij} = o(1/n)$.*

With Lemma 1 and 2, we obtain the main theorem.

Theorem 1 *Assume that (A1) and (A2) hold. Suppose $p < \gamma n$, for a constant $\gamma \in (0,1)$ and $s < \delta p$ for a constant $\delta \in (0,1)$. Let $\lim_{n\to\infty} \alpha_n n^2 = \infty$ and $\alpha_n = o(1/p)$, then, as $n \to \infty$,*

$$\mathbf{P}\Big(\max_{i\in D} |d(i) - (p-1)| > \frac{(1-\delta)p}{4}\Big) = o(1) \tag{4.18}$$

and

$$\mathbf{P}(\max_{j\notin D}|d(j)-s|>\frac{(1-\delta)p}{4})=o(1). \tag{4.19}$$

Theorem 1 essentially says that when $n\to\infty$, with probabilities tending to 1, the degrees of relevant variables are close to $p-1$ and the degrees of irrelevant variables are around s. Moreover, it is easy to show the following result:

Corollary 1 *Suppose the assumptions of Theorem 1 hold, then, as* $n\to\infty$*,*

$$\mathbf{P}\Big(\max_{i_1,i_2\in D}|d(i_1)-d(i_2)|>\frac{(1-\delta)p}{2}\Big)=o(1), \tag{4.20}$$

$$\mathbf{P}\Big(\max_{j_1,j_2\notin D}|d(j_1)-d(j_2)|>\frac{(1-\delta)p}{2}\Big)=o(1), \tag{4.21}$$

$$\mathbf{P}\Big(\min_{i\in D,j\notin D}|d(i)-d(j)|\leq\frac{(1-\delta)p}{2}\Big)=o(1). \tag{4.22}$$

Corollary 1 states that if we use degree as a distance measure, the minimum distance between a relevant variable and an irrelevant variable is asymptotically bigger than the maximum distance between two relevant variable or two irrelevant variables. This implies, if we order the variables by degrees and cut from the biggest gap of the degrees, we can successfully separate relevant variables from irrelevant variables, and thus, achieve *variable selection consistency*. Mathematically, suppose $d(i_1)\leq d(i_2)\leq\ldots\leq d(i_p)$, where $(i_1,i_2,\ldots,i_p)$ is a re-ordering of $(1,2,\ldots,p)$. Let

$$k=\arg\max_l[d(i_l)-d(i_{l-1})] \tag{4.23}$$

and $S = \{i_k, i_{k+1}, \ldots, i_p\}$. We have, as $n \to \infty$,

$$\mathbf{P}(S = D) \to 1. \tag{4.24}$$

Recall that D is the set of relevant variables.

Theorem 1 works for the general case where the number of variables p can either grow to infinity or stay fixed. When p is fixed, and thus s should be fixed too, it is easy to show that, as $n \to \infty$,

$$d(i) \to p - 1, \text{ for } i \in D; \tag{4.25}$$

$$d(j) \to s, \text{ for } j \notin D, \tag{4.26}$$

where the convergence is in expectation. From the proofs of Theorem 1 (Appendix B.3) and Lemma 2 (Appendix B.2), we can see that the above results can be achieved under conditions $\lim_{n\to\infty} \min_{i,j:i\in D \text{ or } j\in D} \Lambda_{ij}/(-\log \alpha_n) = \infty$, $\lim_{n\to\infty} \alpha_n n^2 \to \infty$ and $\alpha_n = o(1)$, which are weaker than those in Theorem 1. In particular, for $\beta_i \neq 0$ or $\beta_j \neq 0$, Λ_{ij} may go to infinity at a rate smaller than $\log n$.

4.2.2 Correlations among Test Statistics

Cutting from the biggest gap of degrees is simple and theoretically intriguing, but it may not perform very well in practice, especially when the sample size n is not very big, whereas the number of variables p is relatively large. We set the significance level of the F-test to

be $\alpha_n = o(1/p)$ to rule out edges among irrelevant variables so that prevent any irrelevant variables from having high degrees. However, as a trade-off, some relevant variables may have fairly low degrees, which could be close to the degrees of irrelevant variables. Besides the substantial size-power dilemma of statistical hypothesis testing, in our situation, this practical challenge is also because of the fact that the test statistics $\{TS_{ij} : j \neq i\}$ are correlated, and thus the Bernoulli random variables $\{A_{ij} = \mathbf{I}(TS_{ij} > C_{\alpha_n}) : j \neq i\}$ are correlated. We discuss the correlations among the test statistics in this section.

First, we look at the test statistics in detail. Recall that

$$TS_{ij} = \frac{Y^T M_{ij} X_{ij} (X_{ij}^T M_{ij} X_{ij})^{-1} X_{ij}^T M_{ij} Y / \sigma^2 / 2}{Y^T (I_n - P(X)) Y / \sigma^2 / (n - p(X))}. \tag{4.27}$$

The denominator is the same for all (i, j) pairs. Aside from the constant $1/2$, the numerator of TS_{ij} is $||P(M_{ij}X_{ij})Y||^2/\sigma^2$, where $P(M_{ij}X_{ij})Y$ is the projection of Y into the space $\mathcal{S}(M_{ij}X_{ij})$. We have shown that $||P(M_{ij}X_{ij})Y||^2/\sigma^2 \sim \chi^2(2, \Lambda_{ij})$. In order to see where the correlation comes from, we decompose the random variable $||P(M_{ij}X_{ij})Y||^2/\sigma^2$. The projection matrix $P(M_{ij}X_{ij})$ can be decomposed as

$$P(M_{ij}X_{ij}) = P(M_i X_i) + P(M_{ij} X_j) \tag{4.28}$$

and

$$P(M_i X_i) P(M_{ij} X_j) = 0. \tag{4.29}$$

The proof can be found at the end of the proof of Lemma 1 in the appendix. Hence,

$$\begin{aligned}\frac{||P(M_{ij}X_{ij})Y||^2}{\sigma^2} &= \frac{||P(M_iX_i)Y||^2}{\sigma^2} + \frac{||P(M_{ij}X_j)Y||^2}{\sigma^2} \\ &:= B_i + C_{ij}, \end{aligned} \tag{4.30}$$

where

$$\begin{aligned}B_i = \frac{||P(M_iX_i)Y||^2}{\sigma^2} &= \frac{Y^TM_iX_i(X_i^TM_iX_i)^{-1}X_i^TM_iY}{\sigma^2} \\ &\sim \chi^2\Big(1, \frac{\beta_i^2X_i^TM_iX_i}{\sigma^2}\Big) \\ &:= \chi^2(1, \Lambda_i^B), \end{aligned} \tag{4.31}$$

because $\mathbf{E}(X_i^TM_iY) = X_i^TM_iX_i\beta_i$ and $\text{var}(X_i^TM_iY) = \sigma^2X_i^TM_iX_i$, and

$$\begin{aligned}C_{ij} = \frac{||P(M_{ij}X_j)Y||^2}{\sigma^2} &= \frac{Y^TM_{ij}X_j(X_j^TM_{ij}X_j)^{-1}X_j^TM_{ij}Y}{\sigma^2} \\ &\sim \chi^2\Big(1, \frac{(X_j^TM_{ij}X_i\beta_i + X_j^TM_{ij}X_j\beta_j)^2}{\sigma^2X_j^TM_{ij}X_j}\Big) \\ &:= \chi^2(1, \Lambda_{ij}^C), \end{aligned} \tag{4.32}$$

because $\mathbf{E}(X_j^TM_{ij}Y) = X_j^TM_{ij}X_i\beta_i + X_j^TM_{ij}X_j\beta_j$ and $\text{var}(X_j^TM_{ij}Y) = \sigma^2X_j^TM_{ij}X_j$. Moreover, B_i and C_{ij} are independent due to (4.29), so

$$\begin{aligned}\frac{||P(M_{ij}X_{ij})Y||^2}{\sigma^2} &= B_i + C_{ij} \\ &\sim \chi^2\Big(2, \Lambda_i^B + \Lambda_{ij}^C\Big), \end{aligned} \tag{4.33}$$

where we have also obtained a decomposition of the noncentrality parameter:

$$\Lambda_{ij} = \Lambda_i^B + \Lambda_{ij}^C. \tag{4.34}$$

Clearly, given a variable i, B_i is a common term for all $\{TS_{ij} : j \neq i\}$, whereas C_{ij} depends on both i and j. Therefore, the correlation between two statistics TS_{ij} and TS_{ik} is induced by the common term B_i and the correlation between C_{ij} and C_{ik}. Next, we investigate how such correlations lead to the issue discussed at the beginning of this section. The effects of correlations on relevant variables and irrelevant variables are explored respectively.

The effect of correlations on relevant variables

We see from the ideal probability matrix (4.14) that, the key to separate a relevant variable i from irrelevant variables is that i has probabilities tending to one to connect to all irrelevant variables. Asymptotically, this is guaranteed by the assumptions and the lemmas. However, when the sample size is finite, the correlations among test statistics may cause problems.

More explicitly, for the relevant variable i, we require that its probabilities to connect to irrelevant variables to be $\alpha_{ij} = 1 - o(1/n)$ for all $j \notin D$. To obtain such a desired asymptotic order of α_{ij}, by Lemma 2, we need the asymptotic order of the noncentrality parameter to be $\lim_{n\to\infty} \Lambda_{ij}/\log n \to \infty$, which is guaranteed by assumptions (A1) and (A2), and Lemma 1. Now we look at the decomposition of Λ_{ij} in detail. Plugging $\beta_i \neq 0$ and

$\beta_j = 0$ into the decomposition (4.34), we have

$$\begin{aligned}\Lambda_{ij} &= \Lambda_i^B + \Lambda_{ij}^C \\ &= \frac{\beta_i^2 X_i^T M_i X_i}{\sigma^2} + \frac{\beta_i^2 (X_j^T M_{ij} X_i)^2}{\sigma^2 X_j^T M_{ij} X_j} \qquad (4.35) \\ &= \frac{\beta_i^2}{\sigma^2} ||M_i X_i||^2 + \frac{\beta_i^2}{\sigma^2} ||P(M_{ij} X_j) X_i||^2. \qquad (4.36)\end{aligned}$$

When the sample size is finite, the common term Λ_i^B, which is the noncentrality parameter of B_i, can be small. On the one hand, the coefficient β_i can be tiny by the design; on the other hand, $||M_i X_i||^2$ is the residual sum of squares from regressing X_i over the remaining columns of X. Given a finite sample size n, $||M_i X_i||^2$ decreases as the number of variables p increases. Even when p is much smaller than n, high sample correlations between X_i and any other columns of X may also result in a small $||M_i X_i||^2$.

Furthermore, the term Λ_i^B being small often implies that all $\{\Lambda_{ij}^C : j \notin D\}$ are small, and thus all $\{\Lambda_{ij} : j \notin D\}$ are small. First, if β_i^2 is small, all $\{\Lambda_{ij}^C : j \notin D\}$ will be small because they have β_i^2 as a common coefficient. Second, note that $||M_i X_i||$ is the L2-norm of the projection of X_i into $\mathcal{S}^{\perp}(X_{-i})$, an $(n - p + 1)$-dimensional space, whereas $||P(M_{ij} X_j) X_i||$ is the L2-norm of the projection of X_i into $\mathcal{S}(M_{ij} X_j)$, a 1-dimensional space. Intuitively, the second term is often much smaller than the first one unless X_i is highly collinear with the vector $M_{ij} X_j$. Therefore, a small $||M_i X_i||$ implies that many $\{||P(M_{ij} X_j) X_i|| : j \notin D\}$ are small.

Overall, when the sample size is finite, for some relevant variable i, the noncentrality parameters $\{\Lambda_{ij} : j \notin D\}$ could be small all together. As a result, the probabilities to

connect to irrelevant variables, i.e., $\{\alpha_{ij} : j \notin D\}$, may be significantly smaller than one simultaneously.

The effect of correlations on irrelevant variables

Consider the aforementioned problem that some relevant variables may have small noncentrality parameters, and thus small probabilities to connect to irrelevant variable. Can we increase the significance level of the F-tests to boost such probabilities? More specifically, can we allow α_n to have a much bigger order than $o(1/p)$? The answer is, because of the correlations of the test statistics, the room to adjust α_n is limited.

For an irrelevant variables i, we have $\beta_i = 0$, and thus $\Lambda_i^B = 0$, so the numerator of TS_{ij} (aside the constant 1/2) is

$$\frac{||P(M_{ij}X_{ij})Y||^2}{\sigma^2} = B_i + C_{ij}, \tag{4.37}$$

with B_i now following $\chi^2(1)$. Again, the common random variable B_i is the main source of the correlations among $\{TS_{ij} : j \neq i\}$. Clearly, if B_i is big, the irrelevant variable i will have considerable probabilities to connect to all the other variables. We show that, if the significance level α_n is set to be much bigger than $o(1/p)$, the chance to have a big B_i over all $i \notin D$ is high. Indeed, in the orthogonal design case, where the random variables $\{B_i : i \notin D\}$ are independent, we can prove the following lemma:

Lemma 3 *Suppose the design matrix is orthogonal. If $s < \delta p$ for a constant $\delta \in (0, 1)$ and, as $n \to \infty$, $n - p \to \infty$ and $\alpha_n^c p \to \infty$ for a constant $c > 2$, then with a probability*

tending to one, there exists an irrelevant variable that connects to all the other variables.

The lemma says that we must not allow α_n to have a bigger order than $1/p^{c_0}$, for any $c_0 < 1/2$. Otherwise, asymptotically, the binary VSN fails for sure in terms of variable selection consistency, because the irrelevant variable who connects to all the other variables will always be selected as a relevant variable. Setting α_n in between $o(1/p)$ and $1/p^{c_0}$ may not result in a complete failure of the VSN. However, some irrelevant variables could connect to many, although not all, other irrelevant variables, which distorts the desired degree distributions, thus increasing difficulty for correct variable selection. Therefore, we should keep $\alpha_n = o(1/p)$.

Remarks

In summary, we must keep $\alpha_n = o(1/p)$ to control false discovery, but as a trade-off, if any relevant variables have small coefficient or are correlated with any other variables, true or irrelevant ones, they could have considerably small probabilities to connect to irrelevant variables. We name such relevant variables as *weak relevant variables*; in contrast, the relevant variables who have probabilities close to one to connect to irrelevant variables are called *strong variables*. In practice, the probability matrix, compared to the ideal one

(4.14), would approximately look like

$$\begin{array}{cccc} & \text{strong} & \text{weak} & \text{irrelevant} \\ \text{strong} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \text{weak} & \mathbf{1} & \boldsymbol{\rho_1} & \boldsymbol{\rho_2} \\ \text{irrelevant} & \mathbf{1} & \boldsymbol{\rho_2}^T & \mathbf{0} \end{array}, \tag{4.38}$$

where $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$ represent block matrices whose elements are mostly in between 0 and 1.

Suppose there are s_1 strong variables and s_2 weak variables. Strong variables still have degrees close to $p-1$. Irrelevant variables now have degrees between s_1 and s_1+s_2. A further difficulty is that weak variables may have degrees distributing all over the interval $(s_1, p-1)$. Moreover, the numbers s_1 and s_2, and entries of $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$ depend on the choice of α_n. Actually, as α_n decreases, some strong variables become weak ones, so s_1 decreases, whereas s_2 increases; meanwhile, entries of $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$ become smaller. As a result, the very weak variables become closer to irrelevant variables in terms of degrees.

Consequently, simply cutting from the biggest gap of degrees may not work very well in practice. In addition, although α_n should be at an order of $o(1/p)$ in theory, it is not obvious what α_n yields the best probability matrix (4.38) in practice. Hence, instead of fixing one value for α_n, we should try various values for it. We will address these issues in Section 4.4.

4.3 A Weighted Variable Selection Network

To build the binary VSN, we set a critical value C_{α_n} and conduct the F-tests. The binary VSN has been shown to have nice theoretical asymptotic properties. However, as discussed in Section 4.2.2, setting the critical value involves trade-offs. In fact, we can skip the thresholding procedure by directly considering a weighted VSN, $\boldsymbol{W} = [W_{ij}]_{p\times p}$, with the (i, j) entry W_{ij} being the numerator of TS_{ij}, the test statistic for the hypothesis test (5.1). The denominator can be ignored because it is the same for all pairs of variables, so that it does not affect any clustering procedure. More explicitly,

$$W_{ij} = Y^T M_{ij} X_{ij} (X_{ij}^T M_{ij} X_{ij})^{-1} X_{ij}^T M_{ij} Y \tag{4.39}$$

and

$$\mathbf{E}(W_{ij}) = \begin{cases} 2\sigma^2 + ||M_{ij}X_i\beta_i + M_{ij}X_j\beta_j||^2, & \text{if } \beta_i \neq 0 \text{ and } \beta_j \neq 0; \\ 2\sigma^2 + \beta_i^2(||M_iX_i||^2 + ||P(M_{ij}X_j)X_i||^2), & \text{if } \beta_i \neq 0 \text{ but } \beta_j = 0; \\ 2\sigma^2, & \text{if } \beta_i = 0 \text{ and } \beta_j = 0. \end{cases} \tag{4.40}$$

As previously discussed, $||P(M_{ij}X_j)X_i||^2$ is usually much smaller than $||M_iX_i||^2$, so its effect is small. Besides the parts $\{||P(M_{ij}X_j)X_i||^2 : i \in D, j \notin D\}$, the weighted VSN has a block structure where each relevant variable belongs to one cluster individually and all irrelevant variables belong to one cluster all together.

For the weighted VSN, cutting from the biggest gap of the degrees does not work well,

and it is hard to establish a nice theoretical result as Theorem 1. The main reason is that the scale of weighted edges is beyond control. In fact, the weighted edges of all relevant variables go to infinity, in expectation, as the sample size goes to infinity. That means, the degrees of all variables go to infinity. Consequently, strong relevant variables usually have explosive degrees while weak relevant variables may have degrees close to irrelevant variables. Therefore, cutting from the biggest gap of degrees usually results in missing weak relevant variables. Although we can not prove any theoretical property for the weighted VSN, it is still useful in practice. We propose a variable selection algorithm for the weighted VSN in Section 4.4 and it is shown to be effective by simulations.

4.4 Variable Selection Network Algorithms

For the binary VSN, we have shown that, with mild conditions, variable selection consistency can be achieved simply by separating the variables (ordered by degrees) at the biggest gap of the degrees (4.13) and selecting the ones with bigger degrees. However, two reasons motivate us to go beyond this easy gap-cutting approach. On the one hand, as discussed in Section 4.2.2, due to the dependence of the test statistics, the block structure of a binary VSN is more complex in practice, so the gap-cutting approach may not yield the best result. On the other hand, for the weighted VSN, the scale of the degrees is unbounded, so that separating from the biggest gap of degrees does not work well. Therefore, we adopt the spectral clustering method described in Section 1.4 to obtain more flexible clustering results of VSNs. We can see, from derivations (1.19) to (1.21), that the spectral clustering approach works as long as $\mathbf{E}(\boldsymbol{A})$ has a block structure, so it can be used for both

the binary VSN and the weighted VSN.

Spectral clustering can cluster variables to any number of clusters. However, how to choose K, the number of clusters, is a challenge in practice. In general, strong variables still form a cluster, yet we do not know how many clusters weak variables and irrelevant variables form. Weak relevant variables can be either close to strong relevant variables or irrelevant variables. Meanwhile, irrelevant variables themselves may also separate to more than one clusters. Therefore, when applying spectral clustering, fixing one K is not the best option.

As discussed at the end of Section 4.2.2, when considering the binary VSN, we should not fix only one significance level α_n either. Consequently, we propose to try a series of α_n and a series of K; conduct variable selection under each setting; and then adopt EBIC (Chen and Chen, 2008) to choose the final model. The EBIC of a submodel $\mathcal{A}$ is calculated as

$$\text{EBIC}(\mathcal{A}) = \log\left\{\frac{1}{n}||\mathbf{I}_n - P(X_{\mathcal{A}})Y||_2^2\right\} + \frac{|\mathcal{A}|}{n}\{\log(n) + 2\log(p)\}. \tag{4.41}$$

We adopt EBIC instead of BIC because our simulations in Section 4.6 mainly deal with $p \geq n$ scenarios.

In this section, we first introduce three variable selection algorithms for the $p < n$ case, and in the next section, we propose a screening procedure for the $p \geq n$ case to reduce the number of variables.

VSN Algorithm 1 Binary VSN with degree gap-cutting

1. Input a series of significance levels, $\alpha_{n1} > \alpha_{n2} > ... > \alpha_{na}$.

2. Under each significance level $\alpha_i, i = 1, 2, ..., a$, construct a binary VSN $\boldsymbol{A}_i$, according to the procedure described in Section 4.2.

3. For each $\boldsymbol{A}_i$, calculate the degrees of all variables and cut from the biggest gap. The variables with higher degrees are recorded as a set $\mathcal{S}_i$.

4. Output the model over $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_a\}$ with the smallest EBIC.

The setting of the significance levels $\alpha_{n1} > \alpha_{n2} > ... > \alpha_{na}$ is quite flexible. We simply let the biggest one $\alpha_{n1} = 0.1$ or 0.05, while the smallest one α_{na} is chosen in a way that, under such a significance level, only very few pairs of variables are significant by the F-test. In regard to a, the length of the series, it should be bigger than the presumed number of relevant variables. For high-dimensional variable selection, the number of relevant variables is usually assumed to be at the same order of $\log(p)$ or $n/\log(p)$, or smaller than $\sqrt{n}$. Making the series too long does not improve the performance. Distances between the numbers do not have to be equal. In our simulation study, $p = 50, 500$ or 1000 and $n = 100$ or 200, so we set the series to be $(e^{-3}, e^{-4}, ...e^{-25})$. We find that moderate changes to the series will not affect the algorithm much.

VSN Algorithm 2 Weighted VSN with spectral clustering

1. Construct a weighted VSN according to the procedure in Section 4.3.

2. Input a series of numbers of clusters, $K_1 < K_2 < ... < K_c$.

3. For each number of cluster $K_i, i = 1, 2, ..., c$, apply the spectral clustering method in

Section 1.4 and obtain clustering results $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_c$, respectively.

4. For each $\mathcal{C}_i, i = 1, 2, ..., c$, set the cluster with the lowest average degree as irrelevant variables and select the rest as relevant variables, denoted as $\mathcal{S}_i$

5. Output the model over $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_s\}$ with the smallest EBIC.

To set the numbers of clusters $K_1 < K_2 < ... < K_c$, we only need to pick the biggest one K_c and set the series to be $2, 3, ..., K_c$. The safe choice of K_c is a number that is bigger than the presumed number of relevant variables, e.g., $\log(p)$, $n/\log(p)$ or $\sqrt{n}$. Practically speaking, when K is big enough, keeping increasing it results in more variables being selected in Step 4, since it is the cluster of irrelevant variables that keeps splitting. Therefore, as far as we observe that the selected model in Step 4 is large enough, we can stop at the corresponding K. In our simulations, we set $K_c = 10$.

VSN Algorithm 3 Binary VSN with spectral clustering

1. Input a series of significance levels, $\alpha_{n1} > \alpha_{n2} > ... > \alpha_{na}$.

2. Under each significance level $\alpha_i, i = 1, 2, ..., a$, construct a binary VSN $\boldsymbol{A}_i$, according to the procedure described in Section 4.2.

3. For each $\boldsymbol{A}_i$, input a series numbers of clusters $K_{i1} < K_{i2} < ... < K_{ic_i}$, apply the spectral clustering method in Section 1.4 and obtain c_i clustering results $\mathcal{C}_{i1}, \mathcal{C}_{i2}, ..., \mathcal{C}_{ic_i}$, respectively.

4. For each $\mathcal{C}_{ij}, i = 1, 2, ..., a, j = 1, 2, ..., c_i$, set the cluster with the lowest average degree as irrelevant variables and select the rest as relevant variables, denoted as $\mathcal{S}_{ij}$.

5. Output the model over $\{\mathcal{S}_{ij} : i = 1, 2, ..., a; j = 1, 2, ..., c_i\}$ with the smallest EBIC.

We can follow the same rules described after Algorithm 1 to set $\alpha_{n1} > \alpha_{n2} > ... > \alpha_{ns}$. For each $i = 1, 2, ..., s$, we also only need to pick a maximum number of clusters K_{ic_i} and set $(K_{i1}, K_{i2}, ..., K_{ic_i}) = (2, 3, ..., K_{ic_i})$. However, the numbers $K_{ic_i}, i = 1, 2, ..., s$, can not be the same. First, for a binary VSN or an adjacency matrix with 0/1 valued entries, the maximum number of possible clusters from the pure mathematical point of view, is generally much smaller than the number of nodes, especially when there are a lot of 0's; in contrast, for a weighted VSN with real-valued entries, the maximum number of possible clusters is usually equal to the number of nodes. In addition, for the binary VSNs constructed under different significance levels, the maximum numbers of possible clusters are usually different. As the significance level decreases, the number of edges in the corresponding binary VSN decreases too. Consequently, the maximum number of possible clusters becomes smaller. Roughly, we have $K_{1c_1} \geq K_{2c_2} \geq ... \geq K_{sc_s}$. We set K_{1c_1} according to the discussion after Algorithm 2, and use it as the universal maximum number of clusters. For each of the rest $i = 2, 3, ..., s$, we attempt one by one from two clusters to K_{1c_1} clusters, and stop when hitting the maximum possible cluster number of the corresponding binary VSN $\boldsymbol{A}_i$.

4.5 An Iterative Group Screening Algorithm

So far, all our discussions are on the $p < n$ scenario. For the $p \geq n$ case, we can not conduct F-test, thus can not construct VSNs. In this section, we propose an iterative group

screening procedure to reduce the number of variables. When facing the $p \geq n$ situation, we first adopt the screening algorithm to reduce the number of variables to $p < n$, and then apply the VSN algorithms in Section 4.4 to conduct variable cleaning.

Screening Algorithm

1. Input a pair of parameters u and r, where u is the number of variables being selected in each iteration and r is the number of iterations. Let the selected variable set $\mathcal{S} = \emptyset$.

2. For each $i \notin \mathcal{S}$, calculate the partial correlation of Y and X_i given $\mathcal{S}$, i.e.,

$$\text{pcor}(Y, X_i|\mathcal{S}) = \frac{\text{cov}\Big(\big(\mathbf{I}_n - P(X_\mathcal{S})\big)Y, \big(\mathbf{I}_n - P(X_\mathcal{S})\big)X_i\Big)}{\sqrt{\text{var}\Big(\big(\mathbf{I}_n - P(X_\mathcal{S})\big)Y\Big) \cdot \text{var}\Big(\big(\mathbf{I}_n - P(X_\mathcal{S})\big)X_i\Big)}}. \tag{4.42}$$

   When $\mathcal{S} = \emptyset$, the partial correlation reduces to the regular correlation.

3. Pick the u variables with the highest partial correlations calculated in the last step, and add them into $\mathcal{S}$.

4. Repeat Step 2 – 3 for $r - 1$ more times.

5. Output $\mathcal{S}$.

The algorithm is inspired by the Iterative Sure Independence Screening (ISIS) (Fan and Lv, 2008). In each iteration, ISIS selects a group of variables by a variable selection method, e.g., SCAD, Lasso, etc, then updates the response as the residuals of regressing Y over the currently selected set of variables, i.e., $Y_{new} = \big(\mathbf{I}_n - P(X_\mathcal{S})\big)Y$. In contrast,

we simply select a fixed number of variables in each iteration by the partial correlations. According to a study of Gaussian graphical model (Buhlmann et al., 2010), partial correlation can better reflect the connection between the response and a variable than regular correlation. Empirically, we also find in our experience that, the partial correlation $\text{pcor}(Y, X_i|\mathcal{S})$ works better than the correlation of the residuals/updated response and X_i, i.e., $\text{cor}\Big(\big(\mathbf{I}_n - P(X_{\mathcal{S}})\big)Y, X_i\Big) = \text{cor}(Y_{new}, X_i)$.

In each iteration, the partial correlation structure of the response and the remaining variables changes according to the selected set $\mathcal{S}$. It is important to select a group of variables in each iteration instead of selecting only one variable because of two reasons: first, in each iteration, a number of variables with top influence are selected instead of the single one with the biggest influence, which eliminates the effects of such a group of variables more thoroughly; second, group selection avoids updating the selected variable set for too many times, which may "distort" the partial correlation structure unexpectedly, especially when the marginal correlation structure of the variables is already complex. On the contrary, given the number of iterations, we clearly should not select too many variables in each iteration, because many irrelevant variables would be chosen.

In our case, the main purpose of the screening procedure is to reduce the number of variables to be smaller than the sample size. We do not require any order of the selected variables, as long as that, by the end of the iterations, all relevant variables are included. In contrast, some advanced screening algorithms, such as the tilted correlation screening (Cho and Fryzlewicz, 2012) and the quantile partial correlation screening (Ma et al., 2016), require an accurate order of variables in the sense that all relevant variables should be selected

ahead of any irrelevant ones.

The choice of two tuning parameters u and r depends on the sample size and the sparsity of the true model. As mentioned previously, for high-dimensional variable selection, one usually assumes the number of relevant variables is at the same order of $\log(p)$ or $n/\log(p)$, or smaller than $\sqrt{n}$. We first choose an overall number for $u \times r$, which should be bigger than the presumed number of relevant variables, and then set u and r. We have discussed above that u should not be too small or too big. Empirically, we suggest setting $u \approx r$. In our simulation study, $p = 50, 500$ or 1000; and $n = 100$ or 200. Hence, we decide to choose 20 variables in the screening step, and set $u = 5$ and $r = 4$. In fact, in our simulations, the numbers of relevant variables vary between 3 to 5, which means we choose about 4 to 7 times as many variables in the screening procedure as the relevant ones. According to the simulations, when the sample size $n = 200$ and the correlations of variables are moderate, varying u and r does not change the results much. When the sample size $n = 100$, or the correlations are strong, varying u and r may lead to mild changes in the results, but does not invalidate any conclusions. Relatively speaking, smaller $u \times r$ (still bigger than the number of relevant variables) usually leads to better final selection results; and conversely, bigger $u \times r$ leads to slightly worse final selection results.

4.6 Simulation Study

In this section, we demonstrate the performance of the variable selection network algorithms by various simulations.

4.6.1 Simulation models

First, we consider two models that are originally from Section 4.2.2 of Fan and Lv (2008) and have been adopted by many following works, for example, Cho and Fryzlewicz (2012) and Ma et al. (2016). The first model considers a difficult situation where one relevant variable is marginally uncorrelated with the response; while in addition to that, the second model further considers a weak signal. Both models are studied under moderate and extremely high correlations, respectively.

(a) A linear regression model:

$$Y = \beta X_1 + \beta X_2 + \beta X_3 - 3\beta X_4\sqrt{\rho} + \epsilon, \tag{4.43}$$

where $\epsilon \sim N(0, \mathbf{I}_n)$ and each row of the p predictors $(X_{i,1}, ..., X_{i,p}), i = 1, 2, ..., n$ are generated independently from a multivariate normal distribution $N_p(\mathbf{0}, \mathbf{\Sigma})$. The covariance matrix $\mathbf{\Sigma}$ is designed in the following way: the (i, j) entry $\Sigma_{ij} = \rho$ for all $i \neq j$, except $\Sigma_{4i} = \Sigma_{j4} = \sqrt{\rho}$, and $\Sigma_{ii} = 1$ for all $i = 1, ..., n$. Such a design implies that X_4 is marginally uncorrelated with Y at the population level.

(b) A more complicated model than (a):

$$Y = \beta X_1 + \beta X_2 + \beta X_3 - 3\beta X_4\sqrt{\rho} + 0.25\beta X_5 + \epsilon, \tag{4.44}$$

where the settings are the same as those of model (a) except $\Sigma_{5i} = \Sigma_{j5} = 0$. The relevant variable X_5 is uncorrelated with any X_i, but it has much smaller coefficients,

and thus is a weak signal.

Following Cho and Fryzlewicz (2012) and Ma et al. (2016), we let $\beta = 2.5$, $\rho = 0.5$ and 0.95. For each model, we run 200 replicates.

The next five models are taken from Narisetty and He (2014). They study scenarios where $p = n$, $p > n$ and $p < n$. Moreover, they also consider weak signals and high correlations.

(c) A linear regression model:

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \epsilon, \tag{4.45}$$

where $\epsilon \sim N_n(0, \mathbf{I}_n)$ and the rows of p predictors $\{(X_{i,1}, ..., X_{i,p}), i = 1, 2, ..., n\}$ are independently generated from a multivariate normal distribution $N_n(\mathbf{0}, \mathbf{\Sigma})$ with $\Sigma_{ij} = 0.25$ for all $i \neq j$ and $\Sigma_{ii} = 1$ for all i. The coefficients of the relevant variables are $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (0.6, 1.2, 1.8, 2.4, 3.0)$.

In the first case, two settings $p = n = 100$ and $p = n = 200$ are considered.

(d) A $p > n$ scenario, where the model settings are the same as those of (c), but $(n, p) = (100, 500)$ and $(n, p) = (200, 1000)$.

(e) The coefficients in (c) are set to $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (0.6, 0.6, 0.6, 0.6, 0.6)$, so that all signals are weak. The other settings are the same as those in (c), while $(n, p) = (100, 500)$.

(f) High correlations, where $\mathbf{\Sigma}$ is set as $\Sigma_{ij} = \Sigma_{ji} = 0.5$ for all $i \leq 5$ and $j \geq 6$,

and $\Sigma_{ij} = \Sigma_{ji} = 0.75$ for all $i \geq 6, j \geq 6$. This means the correlation of one relevant variable and one irrelevant variable is increased to 0.5; the correlation of two irrelevant variables is increased to 0.75; whereas the correlation of any two relevant variables remains 0.25. Again, all the other settings are the same as those in (c), and $(n, p) = (100, 500)$.

(g) A totally different model than the above four ones, which studies the traditional $p < n$ case with $(n, p) = (100, 50)$ and $(200, 50)$. Three of the 50 variables are active ones with coefficients drawn from the uniform distribution $U(0, 3)$. The covariates are multivariate normal with the covariance matrix generated from the Wishart distribution centered at the identity matrix with p degree of freedom.

Same as Narisetty and He (2014), we run 500 replicates for the $p \leq n$ cases and 200 replicates for the $p > n$ cases.

For all models (a) – (g), we first apply the iterative group screening algorithm from Section 4.5 to reduce the number of variables (even for model (g) with $p < n$), and then implement the three variable selection network algorithms from Section 4.4. Regarding tuning parameters, as discussed in Section 4.4 and 4.5, we set the screening parameters $(u, r) = (5, 4)$; the series of significance levels $(\alpha_{n1}, \alpha_{n2}, ..., \alpha_{ns}) = (e^{-3}, e^{-4}, ..., e^{-25})$ for Algorithm 1 and 3; the series of cluster numbers $(K_1, K_2, ..., K_c) = (2, 3, ..., 10)$ for Algorithm 2; and $K_{1c_1} = 10$ for Algorithm 3.

In result tables, the VSN algorithms 1, 2 and 3 are denoted as VSN.bg ("biggest gap"), VSN.sc ("spectral clustering") and VSN.2d, respectively. Algorithm 3 is named VSN.2d because it is "2-dimensional" in the sense that both the significance level and the number

of clusters vary.

For Model (g), we also implement the VSN algorithms directly without the screening procedure, and the algorithms are denoted as VSN0.bg, VSN0.sc and VSN0.2d, respectively.

4.6.2 Simulation results

For Model (a) and (b), Cho and Fryzlewicz (2012) use them to compare the Tilted Correlation Screening methods (TCS) to traditional Forward Selection, Forward Regression (Wang, 2009), LASSO (Tibshirani, 1996), ISIS (Fan and Lv, 2008), PC-Simple algorithm (Buhlmann et al., 2010), MC+ algorithm (Zhang, 2010), SCAD (Fan and Li, 2001) and Forward-Lasso Adaptive Shrinkage (FLASH) (Radchenko and James, 2011); meanwhile, Ma et al. (2016) use them to evaluate the Quantile Partial Correlation Screening (QPCS), the Quantile Tilted Correlation Screening (QTCS) and the Quantile Forward Regression (QFR).

Results for Model (a) and (b) are presented in Table 4.1 and 4.2. The performance of the models are evaluated by the number of false positive selections (FP, the average number of irrelevant variables incorrectly detected as relevant), the number of false negative selections (FN, the average number of relevant variables incorrectly detected as irrelevant), and the sum of FP and FN. Results other than those of the VSN methods are copied from Cho and Fryzlewicz (2012) and Ma et al. (2016). For the quantile screening methods, after the screening procedure, Ma et al. (2016) employ three criteria for the final variable

selection/cleaning: EBIC1, EBIC2 and Lasso. For these two examples, we only report the results of Quantile Screening methods followed by EBIC2, because the results of the other two criteria are much worse and not comparable.

Table 4.1: Simulation results for Model (a)

	$\rho = 0.5$			$\rho = 0.95$		
Method	FP	FN	FP+FN	FP	FN	FP+FN
			$n = 100, p = 1000$			
VSN.bg	0.095	0.005	0.100	0.560	2.005	2.565
VSN.sc	0.130	0.005	0.135	1.270	1.575	2.845
VSN.2d	0.065	0	0.065	0.840	1.600	2.440
TCS1	0.71	0	0.71	0.39	1.43	1.82
TCS2	2.40	0	2.40	0.76	3.64	4.40
FR	22.41	0	22.41	19.84	1.89	21.73
FS	27.86	1.00	28.86	7.14	2.05	9.19
LASSO	58.73	1.00	59.73	28.37	1.54	29.91
ISIS	1.21	3.21	4.42	1.45	3.71	5.16
PCS	2.33	1.65	3.98	1.42	3.58	5.00
MC+	27.94	0.60	28.54	49.58	1.70	51.28
SCAD	111.00	1.00	112.00	46.68	2.07	48.75
FLASH	26.18	1.00	27.18	12.88	1.61	14.49
			$n = 200, p = 1000$			
VSN.bg	0.015	0	0.015	1.380	0.350	1.730
VSN.sc	0.015	0	0.015	1.915	0.265	2.180
VSN.2d	0.015	0	0.015	0.715	0.120	0.835
QPCS.EBIC2	0.145	0	0.145	0.145	0.020	0.165
QTCS.EBIC2	0.740	0	0.740	1.895	0.800	2.695
QFR.EBIC2	1.220	0	1.220	2.680	1.530	4.210

Now we look at the results of Model (a) – Table 4.1. First of all, VSN methods can successfully detect the marginally uncorrelated variable X_4 most of the time. When the correlations among the variables are moderate, e.g., $\rho = 0.5$, all three VSN methods significantly outperform the other methods in terms of FP and FP+FN. VSN.bg and

VSN.sc each incorrectly selects one irrelevant variable over the 200 replicates. When the correlations among the variables are at a very high level, e.g., $\rho = 0.95$, VSN methods still achieve the best FP+FN after TCS1 and QPCS.EBIC2. Although, in terms of FP or FN individually, VSN methods are outperformed by many other methods, they still manage to keep a relatively nice balance between FP and FN. High correlation is a challenge for VSN methods, because it may violate Assumption (A2). As a result, some irrelevant variables, which have high correlations with strong relevant variables, can be detected as relevant ones. Moreover, such strong irrelevant variables can sometimes overshadow weak relevant variables. Finally, VSN.2d, as a "2-dimensional" method, performs better than two "1-dimensional" methods VSN.bg and VSN.sc in most categories.

For Model (b), the results are displayed in Table 4.2. VSN methods can detect the weak signal X_5 most of the time. The results show very similar patterns as those in Table 4.1. Again, when $\rho = 0.5$, VSN methods stand out in FP and FP+FN . Some other methods, such as TCS1 and FR, achieve lower FN values at the cost of higher FP values. When $\rho = 0.95$, VSN methods still remain in the top following TCS1 and QPCS.EBIC2 in terms of FP+FN. Once more, VSN.2d is more superior than VSN.bg and VSN.sc in most categories.

Through Model (c) – (g), Narisetty and He (2014) compare BASAD, Bayesian Variable Selection with Shrinking and Diffusing Priors, to three other Bayesian variable selection methods: piMOM (Johnson and Rossell, 2012), the nonlocal prior method; BCR.Joint (Bondell and Reich, 2012), the Bayesian joint credible region method; and SpikeSlab (Ishwaran and Rao, 2005), the spike and slab method. In addition, they also compare

Table 4.2: Simulation results for Model (b)

	$\rho = 0.5$			$\rho = 0.95$		
Method	FP	FN	FP+FN	FP	FN	FP+FN
			n=100,p=1000			
VSN.bg	0.060	0.195	0.255	0.575	2.780	3.355
VSN.sc	0.090	0.170	0.260	0.825	2.305	3.130
VSN.2d	0.015	0.095	0.110	0.635	2.430	3.065
TCS1	0.85	0.03	0.88	0.05	2.76	2.81
TCS2	3.31	0.11	3.42	0.05	3.96	4.01
FR	30.20	0.01	30.21	26.08	1.75	27.83
FS	29.06	1.15	30.21	4.50	2.32	6.82
LASSO	56.92	1.05	57.97	28.74	1.56	30.30
ISIS	1.23	4.23	5.46	1.03	4.10	5.13
PCS	2.31	2.42	4.73	1.01	3.77	4.78
MC+	32.56	0.79	33.35	35.82	1.86	37.68
SCAD	112.30	1.02	113.30	43.73	2.11	45.84
FLASH	27.04	1.19	28.23	12.78	1.83	14.61
			n=200,p=1000			
VSN.bg	0.010	0	0.010	1.830	0.395	2.225
VSN.sc	0.005	0	0.005	2.020	0.390	2.410
VSN.2d	0.010	0	0.010	0.735	0.115	0.850
QPCS.EBIC2	0.115	0	0.115	0.250	0.035	0.285
QTCS.EBIC2	0.740	0	0.740	1.755	0.925	2.680
QFR.EBIC2	1.390	0.005	1.395	2.555	1.745	4.300

BASAD to three regularization methods: LASSO, Elastic Net (Zou and Hastie, 2005) and SCAD.

The results of Model (c) – (g) are presented in Table 4.3 – 4.7. Four evaluating criteria are reported: $Z = t$, the proportion of exact selection; $Z \supset t$, the proportion of including the true model; FDR, false discovery rate, the proportion of falsely selected irrelevant variables over all irrelevant variables; and MSPE, the average mean squared prediction error based

on n new observations. Results of all the other methods are taken from Narisetty and He (2014).

Table 4.3 displays results of Model (c), which examines the situation of $n = p$, where the five non-zero coefficients vary from 0.6 to 3.0 and the correlations among the variables are mild, i.e., $\Sigma_{ij} = 0.25$ for $i \neq j$. When the sample size is big, e.g., $n = 200$, VSN methods stand out in almost all categories. In particular, they have much higher proportions to select the exact true model and much smaller FDR than the other methods. The only shortcoming is that VSN.bg has a slightly bigger MSPE than BASAD and piMOM. When the sample size is small, e.g., $n = 100$, although VSN methods do not outperform the rest as when $n = 200$, they still achieve high proportions for both $Z = t$ and $Z \supset t$, which are comparable to those of BASAD and piMOM. In contrast, methods such as BASAD.BIC, BCR.Joint, Lasso.BIC, EN.BIC and SCAD.BIC have higher proportions of $Z \supset t$ but smaller proportions of $Z = t$ and relatively high FDR, which means they are likely to overfit the model. In regard to FDR, VSN methods are clearly superior to the other methods. On the one hand, it is due to the screening procedure, which eliminates most irrelevant variables. On the other hand, because of employing EBIC as a criterion, VSN methods tend to select models with small sizes. We will talk more about this issue in the discussion of Model (g). Finally, regarding MSPE, VSN methods are also among the best. Once more, VSN.2d clearly performs better than VSN.bg and VSN.sc.

Model (d) examines the $p > n$ scenario, where the other model settings are the same as Model (c). The results are displayed in Table 4.4. The method piMOM can only deal with the $p \leq n$ case, so can not join the competition. With more variables, all methods are not

Table 4.3: Simulation results for Model (c)

	$(n,p)=(200,200)$				$(n,p)=(100,100)$			
Method	$Z=t$	$Z\supset t$	FDR	MSPE	$Z=t$	$Z\supset t$	FDR	MSPE
VSN.bg	0.986	1.000	<0.001	1.042	0.818	0.862	<0.001	1.092
VSN.sc	0.986	1.000	<0.001	1.025	0.810	0.852	<0.001	1.110
VSN.2d	0.988	1.000	<0.001	1.023	0.876	0.908	<0.001	1.092
BASAD	0.944	1.000	0.009	1.037	0.866	0.954	0.015	1.092
BASAD.BIC	0.090	1.000	0.187	1.087	0.066	0.996	0.256	1.203
piMOM	0.900	1.000	0.018	1.038	0.836	0.982	0.030	1.083
BCR.Joint	0.594	0.994	0.102	1.064	0.442	0.940	0.157	1.165
SpikeSlab	0.008	0.236	0.501	1.530	0.005	0.216	0.502	1.660
Lasso.BIC	0.014	1.000	0.422	1.101	0.010	0.992	0.430	1.195
EN.BIC	0.492	1.000	0.113	1.056	0.398	0.982	0.154	1.134
SCAD.BIC	0.844	1.000	0.029	1.040	0.356	0.990	0.160	1.157

as good as they are for Model (c), yet the patterns of the results are similar to those of Table 4.3. In this case, VSN.2d surpasses BASAD in all categories. Meanwhile, VSN.bg and VSN.sc remain among the best ones. Again, BASAD.BIC, Lasso.BIC, EN.BIC and SCAD.BIC gain high $Z \supset t$ at the cost of overfitting.

Model (e) studies a more challenging situation where all signals are weak, i.e., all five non-zero coefficients are set to be 0.6. From Table 4.5, we see that all methods have difficulty finding the true model. However, compared to the other ones, VSN methods handle the challenge of weak signal well. They outperform the other methods in terms of $Z = t$, FDR and MSPE, while still keep fairly high proportions of $Z \supset t$. Once again, VSN.2d performs better than VSN.bg and VSN.sc. Lasso.BIC achieves the highest proportion of $Z \supset t$ by serious overfitting, with a proportion of $Z = t$ being zero.

Table 4.6 presents the results of Model (f), which examines the scenario of high correlations.

Table 4.4: Simulation results for Model (d)

	$(n,p) = (200, 1000)$				$(n,p) = (100, 500)$			
Method	$Z = t$	$Z \supset t$	FDR	MSPE	$Z = t$	$Z \supset t$	FDR	MSPE
VSN.bg	0.955	0.990	< 0.001	1.025	0.715	0.750	< 0.001	1.134
VSN.sc	0.965	0.985	< 0.001	1.033	0.715	0.740	< 0.001	1.156
VSN.2d	0.975	0.995	< 0.001	1.035	0.770	0.800	< 0.001	1.130
BASAD	0.930	0.950	0.000	1.054	0.730	0.775	0.011	1.130
BASAD.BIC	0.720	0.990	0.046	1.060	0.190	0.915	0.146	1.168
BCR.Joint	0.090	0.250	0.176	1.324	0.070	0.305	0.268	1.592
SpikeSlab	0.000	0.050	0.574	1.933	0.000	0.040	0.626	3.351
Lasso.BIC	0.020	1.000	0.430	1.127	0.005	0.845	0.466	1.280
EN.BIC	0.325	1.000	0.177	1.077	0.135	0.835	0.283	1.223
SCAD.BIC	0.650	1.000	0.091	1.063	0.045	0.980	0.328	1.260

Each pair of relevant variables still has correlation 0.25, but each relevant variable and each irrelevant variable have correlation 0.50, and each pair of irrelevant variables has correlation 0.75. BASAD clearly stands out in terms of $Z = t$, $Z \supset t$ and MSPE, because, according to Narisetty and He (2014), it works similar to the L_0 penalty, so can tolerate high correlations better. VSN methods can be affected by strong correlations, because, as mentioned in the discussion of Model (a), strong correlations may violate Assumption (A2). Despite that, VSN methods still perform well in this situation. Especially, VSN.2d is very close to BASAD. In contrast, most of the other methods are broken by the high correlations.

Combining these results with those of Model (a) and (b), we conclude that VSN methods can accommodate high correlations relatively well. They have no problem handling moderate level correlations, e.g., around 0.5. When the correlation is extremely high, e.g., up to 0.75 or 0.95, VSN methods are still competitive with, although not as good as, the

Table 4.5: Simulation results for Model (e)

	$(n, p) = (100, 500)$			
Method	$Z = t$	$Z \supset t$	FDR	MSPE
VSN.bg	0.265	0.320	< 0.001	1.496
VSN.sc	0.275	0.340	< 0.001	1.495
VSN.2d	0.310	0.350	< 0.001	1.480
BASAD	0.185	0.195	0.066	2.319
BASAD.BIC	0.160	0.375	0.193	1.521
BCR.Joint	0.030	0.315	0.447	1.501
SpikeSlab	0.000	0.000	0.857	2.466
Lasso.BIC	0.000	0.520	0.561	1.555
EN.BIC	0.040	0.345	0.478	1.552
SCAD.BIC	0.045	0.340	0.464	1.561

best-performing methods such as TCS1, QPCS.EBIC2 and BASAD.

Finally, Model (g) studies the traditional $p < n$ situation and the results are shown in Table 4.7. For VSN methods, we report two sets of results, with and without the screening procedure, respectively. The ones without screening are denoted by VSN0. With screening, VSN methods outperform all the other methods in almost all categories. Without screening, VSN0.bg and VSN0.2d remain competitive with the best-performed methods BASAD and piMOM in terms of $Z = t$, $Z \supset t$ and FDR. However, the MSPE values of VSN0.bg and VSN0.2d are much worse than the rest. This is because, compared to the other methods, VSN0.bg and VSN0.2d are more likely to choose undersized model so that they tend to miss more relevant variables. Although the average false negative number is not reported in Table 4.7, we can see it being big from the small FDR. Since Model (g) only has three relevant variables, missing some of them can result in huge prediction errors. Now we look at the results of VSN0.sc. Comparing to VSN.sc, now without the pre-screening,

Table 4.6: Simulation results for Model (f)

	$(n, p) = (100, 500)$			
Method	$Z = t$	$Z \supset t$	FDR	MSPE
VSN.bg	0.295	0.350	< 0.001	1.339
VSN.sc	0.295	0.355	< 0.001	1.316
VSN.2d	0.425	0.455	< 0.001	1.278
BASAD	0.505	0.530	0.012	1.190
BASAD.BIC	0.165	0.815	0.179	1.210
BCR.Joint	0.000	0.000	0.515	2.212
SpikeSlab	0.000	0.000	0.995	10.297
Lasso.BIC	0.000	0.015	0.869	8.579
EN.BIC	0.000	0.000	0.898	8.360
SCAD.BIC	0.000	0.000	0.899	8.739

the performance of VSN0.sc drops dramatically in regard to $Z = t$, $Z \supset t$ and MSPE. Recall that Algorithm 2 works on the original weighted network of test statistics without any thresholding. When all variables are included in the network, what usually happens is that some extremely weak irrelevant variables would form the cluster with the lowest degree. This implies, instead of the relevant variables standing out, it is more likely that the extremely weak variables step back. Due to the design, Algorithm 2 selects all variables other than the cluster with lowest degree as relevant variables. Therefore, without the pre-screening procedure, Algorithm 2, i.e., VSN0.sc, tends to overfit the model. This can also be seen from its high FDR. Overall, this simulation shows that, for the $p < n$ case, Algorithm 1 (VSN0.bg) and 3 (VSN0.2d) perform very well without the screening procedure, whereas Algorithm 2 (VSN0.sc) has problems. The screening can significantly improve the performance of Algorithm 2, and boost the performance of Algorithm 1 and 3 as well.

Table 4.7: Simulation results for Model (g)

	$(n,p)=(200,50)$				$(n,p)=(100,50)$			
Method	$Z=t$	$Z\supset t$	FDR	MSPE	$Z=t$	$Z\supset t$	FDR	MSPE
VSN.bg	0.938	0.948	<0.001	1.017	0.924	0.940	<0.001	1.039
VSN.sc	0.820	0.900	0.006	1.029	0.802	0.894	0.006	1.071
VSN.2d	0.938	0.948	<0.001	1.018	0.928	0.944	<0.001	1.033
VSN0.bg	0.790	0.796	<0.001	1.740	0.672	0.678	<0.001	2.902
VSN0.sc	0.102	0.838	0.220	1.180	0.094	0.736	0.200	2.039
VSN0.2d	0.886	0.910	0.002	1.025	0.808	0.870	0.004	1.117
BASAD	0.738	0.784	0.017	1.029	0.654	0.714	0.026	1.086
BASAD.BIC	0.338	0.842	0.193	1.055	0.208	0.778	0.267	1.151
piMOM	0.694	0.740	0.020	1.036	0.656	0.708	0.021	1.066
BCR.Joint	0.484	0.770	0.133	1.045	0.336	0.650	0.216	1.124
SpikeSlab	0.038	0.900	0.629	1.121	0.064	0.846	0.567	1.226
Lasso.BIC	0.082	0.752	0.378	1.059	0.076	0.744	0.397	1.152
EN.BIC	0.428	0.748	0.165	1.039	0.378	0.742	0.194	1.110
SCAD.BIC	0.358	0.812	0.193	1.046	0.186	0.772	0.284	1.147

In summary, through the simulation study, VSN methods have been proven to be powerful. They outperform many state-of-the-art methods, and are very competitive with the latest and advanced screening based approaches and Bayesian variable selection approaches. VSN methods are able to accommodate high-dimensionality, weak signals and moderate to high level correlations. They are superior in terms of choosing the exact true model, and they always maintain a nice balance between false positive and false negative selections. Occasionally, some VSN methods may underfit the model.

4.7 A Real Data Application

In this section, we apply the VSN methods to the *Communities and Crime Unnormalized Data Set* from the UCI Machine Learning Repository. The data records information about communities in the U.S., which combines the socio-economic data from the 1990 U.S. Census, the law enforcement data from the 1990 U.S. Law Enforcement Management and Admin Stats survey, and crime data from the 1995 U.S. FBI UCR. Details of this data set can be found on the web page:

https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized.

The original data set consists of 2215 rows and 147 columns, including 125 predictive variables, 4 non-predictive variables and 18 potential response variables. First, we delete columns with 5 or more missing values and then delete rows with missing values. Second, we delete two columns "OwnOccQrange" and "RentQrange" which turn out to be linearly dependent on other variables. In the end, the final data contains 2206 rows and 113 columns, including 100 numeric predictive variables, 1 categorical predictive variable (state), 2 non-predictive variables and 10 potential response variables.

The goal of this application is to investigate what attributes of communities are statistically related to numbers of murders in communities. We use "murdPerPop", number of murders per 100K population, as our response variable and the 100 numeric predictive variables as predictors.

Although this is a $p < n$ scenario, we still apply the group iterative screening algorithm to the variables first, because as shown in simulation Model (g), the screening procedure can

improve the performance of the VSN methods. The sample size $n = 2215$, so we pick the parameters for the screening algorithm to be $u = r = 7$, such that $u \times r \approx \sqrt{n}$. Then we apply the three VSN methods to the 49 screened variables. We set the series of significance levels to be $(e^{-3}, e^{-4}, \ldots, e^{-35})$ and the series of numbers of clusters to be $(2, 3, 4, \ldots, 25)$. The two series are longer than those used in the simulation study because we now have more variables and more samples.

The variables selected by the three VSN methods are presented in Table 4.8, respectively. The variables are ordered by their degrees in the weighted VSN (Section 4.3), i.e., $\sum_{j=1}^{p} W_{ij}$, from largest to smallest. For the sake of a clearer view with smaller magnitude, we present $\sum_{j=1}^{p} TS_{ij}$ (sum of test statistics), which is a rescaling of the weighted degree (divided by a common constant), in the degree column.

Over all 100 variables, VSN.bg selects 15; VSN.sc selects 12 and VSN.2d selects 7. All three methods choose the top same 6 variables, which means they are very important. For the other variables, VSN.bg and VSN.sc have some overlap, while VSN.2d identifies just one additional variable. We can see that VSN.sc and VSN.2d select some variables with smaller degrees but exclude some variables with larger degrees. This property may be due to the ability of spectral clustering to consider more complex network structures beside degrees. One interesting result is that, although VSN.2d selects fewest variables, it discovers "percentage of males who are divorced" which is not in the lists of VSN.bg and VSN.sc.

Table 4.8: Variables selected for the communities and crime data by the VSN methods.

Attributes of communities	Degrees	VSN.bg	VSN.sc	VSN.2d
Percentage of vacant housing boarded up	713.13	X	X	X
Percentage of African American	335.28	X	X	X
Percentage of Caucasian	281.71	X	X	X
Number of people under poverty	251.98	X	X	X
Population per square mile	215.41	X	X	X
Percentage of people not speaking English well	191.10	X	X	X
Percentage of officers assigned to drug units	189.51	X		
Percentage of households with wage or salary	180.80	X		
Number of vacant households	179.76	X		
Number of homeless people	176.83	X		
Average number of persons per owner occupied household	174.11	X	X	
Percentage of population between 12-29 in age	172.94	X	X	
Percentage of vacant housing over 6 months	169.44	X	X	
Percentage of housing without plumbing facilities	166.19	X		
Average number of people per family	162.44	X	X	
Percentage of people who speak only English	147.04		X	
Percentage of males who are divorced	137.41			X
Percentage of housing with less than 3 bedrooms	121.84		X	

4.8 Summary

In this chapter, we have proposed Variable Selection Networks, a novel variable selection method within the framework of Variable Selection Ensembles (Xin and Zhu, 2012). By considering the ensemble of all pairs of variables, we construct variable selection networks (VSN). For the $p < n$ case, edges of a VSN are determined by the F-tests for the binary VSN or the F-test statistics for the weighted VSN. We show that, for a VSN, the edges' distributions have a block structure. Furthermore, for the binary VSN, we establish theoretical properties such as the asymptotic degree distributions and variable selection consistency, and we also investigate the correlations among the edges. Incorporating both theoreti-

cal and practical perspectives, we propose three VSN algorithms for the $p < n$ scenario. To handle the high-dimensional $p \geq n$ scenario, we introduce an iterative group screening algorithm, which can reduce the number of variables while keep relevant variables. Simulations show that the VSN algorithms' performance is outstanding when compared to many state-of-the-art methods, latest screening based methods and various Bayesian variable selection methods. In the end, the VSN methods are applied to a real data example.

Chapter 5

Summary and Future Research

5.1 Summary of the Thesis

This thesis considers generalizations and applications of the Stochastic Block Model.

In Chapter 2, we generalize the standard SBM to a Continuous-time Stochastic Block Model (CSBM) to conduct community detection for transactional networks. The CSBM differs from many existing methods in that it considers time to be continuous. Transactions between each pair of nodes are modeled as an inhomogeneous Poisson process, with the rate function depending only on the community labels of the two nodes. We use cubic B-splines to model the rate functions. An EM algorithm is developed to fit the CSBM. In the E-step, due to the complexity of the model, Gibbs sampling is adopted to generate samples from the conditional distribution and estimate the conditional means. A simple simulation example shows that the CSBM and the EM algorithm work well.

In Chapter 3, we develop a multistate Continuous-time Stochastic Block Model and apply it to basketball games. First of all, we provide a new perspective that basketball games can be analyzed as transactional networks, with players being nodes and ball passes among players being transactions. Additionally, initial actions of basketball plays and play outcomes are considered as special nodes of basketball networks. A multistate CSBM is proposed to cluster players based on their styles of handling the ball. In particular, each basketball play, in which the ball travels through different type of nodes, is modeled as an inhomogeneous continuous-time Markov chain. The transition rate functions are governed by the players' underlying cluster memberships. We adopt B-splines to model the rate functions and develop an EM^+ algorithm to estimate the model parameters. The multistate CSBM is applied to a number of NBA games between the 2011-12 Miami Heat and Boston Celtics and between the 2014-15 Cleveland Cavaliers and Golden State Warriors. The clustering results are consistent with common understanding of the players in these games. Moreover, the estimated transition rate functions and transition probabilities reveal insightful details in offensive strategies of these teams. Overall, the multistate CSBM is shown to be of great practical value in clustering and evaluating basketball players. Although we have been focusing on basketball in the thesis, the multistate CSBM provides a general framework for modeling any transactional network similar to the basketball network, which has one object moving among the nodes.

In Chapter 4, we propose Variable Selection Networks, a novel variable selection method within the framework of Variable Selection Ensemble. This method is fundamentally different than variable selection methods such as penalized likelihood methods, variable screening methods and Bayesian variable selection methods. Variable selection networks

(VSNs) are constructed by considering all pairs of variables, with each variable being a node and each edge being a measure of the importance of the corresponding pair of nodes. Such VSN are shown to have block structures, so that techniques of the Stochastic Block Model can be utilized to analyze them. For the $p < n$ case, we first consider a binary VSN, with the (i, j) edge being determined by the F-test for the hypothesis test $\mathbf{H}_0 : \beta_i = \beta_j = 0$ vs. $\mathbf{H}_1 : \beta_i \neq 0$ or $\beta_j \neq 0$, i.e., the edge is 1 if the hypothesis is rejected or 0 otherwise. We establish that, under mild conditions and with a proper significance level of the F-test, variable selection consistency can be achieved by simply cutting from the biggest gap of node degrees of the VSN and selecting variables with higher degrees. Despite the sound theoretical properties, we demonstrate that the correlations among test statistics cause practical challenges. Beside the binary VSN, a weighted VSN is constructed as well. To overcome the practical challenges and to utilize the weighted VSN, we propose three VSN algorithms which incorporate the EBIC by Chen and Chen (2008) for high-dimensional variable selection and the spectral clustering algorithm for the Stochastic Block Model by Lei and Rinaldo (2015). For the $p > n$ scenario, we propose an iterative group screening algorithm to reduce the number of variables while retaining relevant variables. Essentially, when $p > n$, we first apply the screening algorithm to reduce the number of variables to be smaller than n , and then apply VSN algorithms for variable selection. Comprehensive simulations illustrate the performances of VSN algorithms under difficult situations such as extremely high correlations among variables, weak signals, variables marginally uncorrelated with responses, etc. Overall, the VSN algorithms are shown to be able to accommodate these tough situations. They perform outstandingly compared to many state-of-the-art methods, latest screening based methods and various Bayesian

variable selection methods. In particular, they are superior in terms of choosing the exact true model and always maintain a nice balance between false positive and false negative selections.

5.2 Future Research

5.2.1 Continuous-time Stochastic Block Models

For the Stochastic Block Model, an open problem is to evaluate goodness-of-fit of the model, including determining the number of communities. For generic networks, considerable attention has been devoted to this problem very recently. Wang and Bickel (2016) consider an approach based on the log-likelihood ratio statistic and propose a penalized likelihood model selection criterion that is asymptotically consistent in terms of selecting the true number of clusters. Saldana et al. (2016) propose a composite likelihood BIC criterion. Lei (2016) investigates the residual matrix obtained by substracting the estimated block mean effect from the original adjacency matrix and proposed a goodness-of-fit test based on the largest singular value of the residual matrix. Chen and Lei (2016) develop a network cross-validation method to determine the number of communities. Naturally, we would like to develop a goodness-of-fit test or criterion for Continuous-time Stochastic Block Models. One possible direction is to learn from likelihood based methods such as Wang and Bickel (2016) and Saldana et al. (2016).

In addition, we will consider more complicated parametrizations of rate functions. Network

statistics can be used as covariates when modeling the rate functions, for example, DuBois et al. (2013) and Vu et al. (2011). External information can also be incorporated into the rate functions. Moreover, as in event history analysis (Cook and Lawless, 2007), we can consider intensity functions instead of rate functions, with historical information being taken into account.

5.2.2 Basketball Networks

We need to find more and richer basketball data, and construct more complex multistate CSBMs. The lack of data restrains us from applying the model to more games. The manually collected data contains limited information. If we had access to optical tracking data (Cervone et al., 2016), we could incorporate more details into the rate functions, for instance, players' spatial positions, defensive levels, etc. We believe that, with richer data, the CSBM framework would be of even greater practical value in clustering and evaluating basketball players.

Intuitively, players have individual features. For example, even being clustered in the same cluster, a star player should be different with a regular player. Hence, we may add individual level parameters, such as in the Degree Corrected Stochastic Blockmodel (Karrer and Newman, 2011).

An interesting problem is to detect whether there is a change of strategy for a team in a game or between games (when facing different opponents). In Section 3.3.3, for each team, we analyzed the two games separately and, by comparing the results, we found that

the Warriors changed strategy while the Cavaliers did not change much. Following this direction, we may develop some statistical test to test changes. A difficulty is that, because of the potential cluster or player differences, the model for two games together and the combined model from fitting two games separately are not nested models.

5.2.3 Variable Selection Networks

First, in a broad sense, the Variable Selection Networks (VSN) framework can be viewed as a "pairwise variable screening" procedure, because the VSN considers all pairs of variables. In contrast, the Sure Independence Screening (SIS)(Fan and Lv, 2008) or marginal screening considers all variables individually. Following this direction, the VSN can be seen as an extension of the SIS. By far, all screening methods only focus on individual screening. Further exploring the concept of pairwise variable screening is part of the future work.

Second, for the binary VSN, we prove that, with mild conditions, variable selection consistency can be achieved simply by separating the variables (ordered by degrees) from the biggest gap of the degrees (4.13) and selecting the ones with bigger degrees. The corresponding Algorithm 1 shows convincing results in simulations. However, such a simple gap-cutting algorithm is not very flexible when dealing with more complex binary VSNs in practice. Moreover, it is not able to handle the weighted VSN, whose scale of the degrees is unbounded. Therefore, we apply a spectral clustering method to VSNs. Simulations show that Algorithm 2, weighted VSN with spectral clustering, is as powerful as Algorithm 1; meanwhile, Algorithm 3, binary VSN with spectral clustering, is clearly superior to

Algorithm 1. Establishing theoretical properties of the spectral clustering for VSNs, i.e., Algorithm 2 and 3, is of great interest and worth investigation in the future.

Lei and Rinaldo (2015) prove the clustering consistency of the spectral clustering for the Stochastic Block Model, given K, the number of clusters, is known. More explicitly, for the spectral clustering method in Section 1.4, to assure the performance of the algorithm, or to have any clustering consistency, it is required that the distance between $\hat{\boldsymbol{U}}$ and $\boldsymbol{U}$ not be large. Indeed, Lei and Rinaldo (2015) show that

$$||\hat{\boldsymbol{U}} - \boldsymbol{U}||_F \leq \frac{2\sqrt{2K}}{\gamma_p}||\boldsymbol{A} - \boldsymbol{Q}||, \tag{5.1}$$

where $||\cdot||_F$ and $||\cdot||$ denote the Frobenius norm and the spectral norm of a matrix, respectively; γ_p is the smallest nonzero eigenvalue (in absolute value) of $\boldsymbol{Q}$. In addition, with extra settings of the Stochastic Block Model: A_{ij}'s are Bernoulli random variables and edges are independent given cluster labels, the authors prove that assuming $p \max_{kl} B_{kl} = d_n \geq c_0 \log p$, there exists a constant $C = C(c_0)$, such that

$$||\boldsymbol{A} - \boldsymbol{Q}|| \leq C\sqrt{d_p} \tag{5.2}$$

with probability at least $1 - n^{-1}$.

Finally, they prove the result

$$\sum_{k=1}^{n} \frac{|S_k|}{p_k} \leq c^{-1}(2+\delta)\frac{K d_p}{\gamma_p^2}, \tag{5.3}$$

where $|S_k|$ is the number of mis-clustered nodes in cluster k; and δ is from the $(1+\delta)$-approximate k-means clustering (Kumar et al., 2004). Lei and Rinaldo (2015) use the $(1+\delta)$-approximate because it can find an approximate solution which is within $(1+\delta)$ fraction of the optimal solution in linear time, whereas finding the exact optimal solution of k-means is NP-hard.

The inequality (5.3) provides an upper bound for the sum of fractions of mis-clustered nodes in all clusters. Lei and Rinaldo (2015) argue that in many cases, the term on the right hand side goes to zero as $p \to \infty$, so the fractions of mis-clustered nodes vanish. In other words, clustering consistency is achieved on a proportion level.

For the binary VSN, proving cluster consistency is much harder. First of all, we obviously need some lower bounds on the elements of $\boldsymbol{\rho_2}$ in (4.38) to make weak relevant variables separable from irrelevant ones. Second, the proof of (5.2) relies on the independence of the elements of $\boldsymbol{A}$, but as discussed in Section 4.2.2, they are not independent for the VSN. Constraints on the correlations are needed in order to bound $||\boldsymbol{A} - \boldsymbol{Q}||$. Finally, we need homogeneity assumptions on each row of $\boldsymbol{\rho_2}$ in (4.38). We do not want irrelevant variables to form too many clusters. Otherwise, the number of clusters K becomes large, so the bound in the inequality (5.1) becomes loose.

Theoretical proof is even more challenging for the weighted VSN. It is harder to find a bound as (5.2), since the weighted edges are unbounded. Note that by far all theoretical results regarding the Stochastic Block Model are built upon binary networks.

Despite the difficulties just outlined, unlike spectral clustering for the Stochastic Block Model, which hopes to exactly recover all clusters, our goal is slightly easier, where we

only need to separate relevant variables from irrelevant ones. We succeed as long as the irrelevant variables are clustered together so that they can be separated from relevant variables. Hence, we may not need a result as comprehensive as (5.3). In this sense, achieving variable selection consistency results may still be possible.

References

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

Peter J. Bickel and Aiyou Chen. A nonparametric view of network models and newman-girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106 (50):21068–21073, 2009.

Howard Bondell and Brian Reich. Consistent high-dimensional Bayesian variable selection via penalized credible regions. *Journal of the American Statistical Association*, 107(500): 1610–1624, 2012.

Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37 (4):373–384, 1995.

Peter Buhlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer, Verlag Berlin Heidelberg, 2011.

Peter Buhlmann, Markus Kalisch, and Marloes H. Maathuis. Variable selection in high-dimensional linear models: partially faithful distributions and the PC-simple algorithm. *Biometrika*, 97(2):261–278, 2010.

Emmanuel Candes and Terence Tao. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

Daniel Cervone, Alex D'Amour, Luke Bornn, and Kirk Goldsberry. A multireolution stochastic process model for predicting basketball possession outcomes. *Journal of the American Statistical Association*, 111(514):585–599, 2016.

Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

KeHui Chen and Jing Lei. Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, to appear, 2016.

Louis H.Y. Chen and Qi-Man Shao. A non-uniform Berry-Esseen bound via Stein's method. *Probability Theory and Related Fields*, 120:236–254, 2001.

Haeran Cho and Piotr Fryzlewicz. High dimensional variable selection via tilting. *Journal of the Royal Statistical Society: Series B*, 74(3):593–622, 2012.

David S. Choi, Patrick J. Wolfe, and Edoardo M. Airoldi. Stochastic blockmodels with a

growing number of classes. *Biometrika*, 99(2):273–284, 2012.

Richard J. Cook and Jerald F. Lawless. *The stiatistical analysis of recurrent events*. Springer, New York, NY, 2007.

James A. Davis. Clustering and hierarchy in interpersonal relations: Testing two graph theoretical models on 742 sociomatrices. *American Sociological Review*, 35(5):843–851, 1970.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

Christopher DuBois, Carter T. Butts, and Padhraic Smyth. Stochastic blockmodeling of relational event dynamics. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Paul Erdös and Aflréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

Jianqing Fan and Runzhe Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensioanal feature space. *Journal of the Royal Statistical Society: Series B*, 70(5):849–911, 2008.

Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional

feature space. *Statistica Sinica*, 20(1):101–148, 2010.

Jennifer H. Fewell, Dieter Armbruster, John Ingraham, Alexander Petersen, and James S. Waters. Basketball teams as strategic networks. *PLoS ONE*, 7(11):849–911, 2012.

Alexander Franks, Andrew Miller, Luke Bornn, and Kirk Goldsberry. Characterizing the spatial structure of defensive skill in professional basketball. *The Annals of Applied Statistics*, 9(1):94–121, 2015.

Edgar Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

Qirong Ho, Le Song, and Eric P. Xing. Evolving cluster mixed-membership blockmodel for time-varying networks. *Proceedings of the 14th International Conference on Artifical Intelligence and Statistics (AISTATS)*, 2011.

Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Hemant Ishwaran and J. Sunil Rao. Spike and slab variable selection frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

Jiashun Jin. Fast community detection by score. *The Annals of Statistics*, 43(1):57–89, 2015.

Jiashun Jin, Cun-Hui Zhang, and Qi Zhang. Optimality of graphlet screening in high dimensional variable selection. *Journal of Machine Learning Research*, 15:2723–2772, 2014.

Valen E. Johnson and David Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107, 2011.

B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

Eric D. Kolaczyk. *Statistical Analysis of Network Data*. Springer, New York, NY, 2009.

Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. *In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 454–462, 2004.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Jing Lei. A goodness-of-fit test for stochastic block models. *The Annals of Statistics*, 44 (1):401–424, 2016.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

Jure Leskovec and Eric Horvitz. Planetary-scale views on an instant messaging network. *arXiv:0803.0939*, 2008.

Dean Lusher, Johan Koskinen, and Garry Robins. *Exponential Random Graph Models for Social Networks*. Cambridge University Press, New York, NY, 2012.

Shujie Ma, Runze Li, and Chih-Ling Tsai. Variable screening via quantile partial correlation. *The Annals of Statistics*, Published online, 2016.

Nicolai Meinshausen and Peter Buhlmann. Stability selection. *Journal of the Royal Statistical Society*, 72(4):417–473, 2010.

R. Michel. On the constant in the nonuniform version of the Berry-Esseen theorem. *Z. Wahrsch. Verw. Gebiete*, 55:109–117, 1981.

Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. *Proceedings of the* 31^{th} *International Conference on Machine Learning*, 2014.

Naveen NaiDu Narisetty and Xuming He. Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817, 2014.

Mark E.J. Newman. Mixing patterns in networks. *Physical Review E*, 67:026126, 2003.

Bob O'Hara and Mikko J. Sillanpaa. A review of Baysian variable selection methods: what, how and which. *Bayesian Analysis*, 4(1):85–118, 2010.

Dean Oliver. *Basketball On Paper: Rules and Tools for Performance Analysis*. Potomac Books, Inc., Dulles, Virginia, 2004.

Derek De Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306, 1976.

Peter Radchenko and Gareth M. James. Improved variable selection with forward-lasso adaptive shrinkage. *The Annals of Applied Statistics*, 5(1):427–448, 2011.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Diego Franco Saldana, Yi Yu, and Yang Feng. How many communities are there? *Journal of Computational and Graphical Statistics*, to appear, 2016.

Michael Schweinberger and Tom A. B. Snijders. Settings in social networks: A measurement model. *Sociological Methodology*, 33(1), 2003.

Mahdi Shafiei and Hugh Chipman. Mixed-membership stochastic block-models for transactional networks. *Proceedings of the International Conference on Data Mining*, 2010.

Stephen M. Shea and Christopher E. Baker. *Basketball Analytics: Objective and Efficient Strategies for Understanding How Teams Win*. Advanced Metrics, LLC, Lake St. Louis, MO, 2013.

Tom A. B Snijders. Stochastic actor-oriented models for network change. *Jouranl of Mathematical Sociology*, 21:149–172, 1996.

Tom A. B Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395, 2001.

Tom A. B. Snijders. Statistical models for social networks. *Annual Review of Sociology*, 37:131–153, 2011.

Tom A. B. Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

Rob Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

Duy Q. Vu, Arthur U. Asuncion, David R. Hunter, and Padhraic Smyth. Continuous-time regression models for longitudinal networks. *Advances in Neural Information Processing Systems*, 2011.

Hansheng Wang. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104(488):1512–1524, 2009.

Yuchung J. Wang and George Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

Y.X. Rachel Wang and Peter Bickel. Likelihood-based model selection for stochastic block models. *arXiv:1502.02069v3*, 2016.

Larry Wasserman and Kathryn Roeder. High-dimensional variable selection. *The Annals of Statistics*, 37(5A):2178–2201, 2009.

Duncan Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Lu Xin and Mu Zhu. Stochastic stepwise ensembles for variable selection. *Journal of Computational and Graphical Statistics*, 21(2):275–294, 2012.

Lu Xin, Mu Zhu, and Hugh Chipman. A continuous-time stochastic block model for basketball networks. *The Annals of Applied Statistics*, accepted, 2016.

K. S. Xu and A. O. Hero. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):552–562, 2014.

Cunhui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

Cunhui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stoachastic block models. *The Annals of Statistics*, 40 (4):2266–2292, 2012.

Mu Zhu and Hugh Chipman. Darvinian evolution in parallel universes:a parallel genetic algorithm for variable selection. *Technometrics*, 48(4):491–502, 2006.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.

Appendix A

Some Details for the EM Algorithm in Section 3.2.2

A.1 The Conditional Expectation $\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z})|\mathbf{T}; \Theta^*]$

First, by (3.1), we have

$$
\begin{aligned}
&\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z})|\mathbf{T}; \Theta^*] \qquad (A.1)\\
&= \mathbf{E}[\log \mathcal{L}(\mathbf{T}|\mathbf{Z}) + \log \mathcal{L}(\mathbf{Z})|\mathbf{T}; \Theta^*]\\
&= \mathbf{E}\Big[\sum_{s\in\mathcal{S}}\sum_{i=1}^{n} \log \mathcal{L}^{I}(\mathbf{T}_{si}|\mathbf{Z}) + \sum_{1\le i\ne j\le n} \log \mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{Z})\\
&\quad + \sum_{i=1}^{n} \log \mathcal{L}^{P_2}(\mathbf{T}_{i}|\mathbf{Z}) + \sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \log \mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z}) + \log \mathcal{L}(\mathbf{Z})\Big|\mathbf{T}; \Theta^*\Big]
\end{aligned}
$$

Now, we plug in (3.15), (3.16), (3.17), (3.18) and (3.19), and the respective terms in (A.1) are as follows. The $\mathcal{L}^I$ part is equal to

$$\begin{aligned}
&\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\mathbf{E}\Big[\log\mathcal{L}^I(\mathbf{T}_{si}|\mathbf{Z})\Big|\mathbf{T};\Theta^*\Big] \qquad (A.2)\\
&\quad=\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\mathbf{E}\Big[\log\prod_{k=1}^{K}\Big(\prod_{h=1}^{m_{si}}\big(P_{sk}\cdot\frac{1}{G_k^{sih}}\big)\Big)^{z_{ik}}\Big|\mathbf{T};\Theta^*\Big]\\
&\quad=\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbf{E}\Big[z_{ik}\cdot\sum_{h=1}^{m_{si}}\big(\log P_{sk}-\log G_k^{sih}\big)\Big|\mathbf{T};\Theta^*\Big]\\
&\quad=\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{si}\log P_{sk}\Big)\\
&\qquad\qquad-\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbf{E}\Big[z_{ik}\cdot\sum_{h=1}^{m_{si}}\log G_k^{sih}\Big|\mathbf{T};\Theta^*\Big].
\end{aligned}$$

The $\mathcal{L}^{P_1}$ part is equal to

$$\begin{aligned}
&\sum_{1\le i\neq j\le n}\mathbf{E}\Big[\log\mathcal{L}^{P_1}(\mathbf{T}_{ij}|\mathbf{Z})\Big|\mathbf{T};\Theta^*\Big] \qquad (A.3)\\
&=\sum_{1\le i\neq j\le n}\mathbf{E}\Big[\log\prod_{k=1}^{K}\prod_{l=1}^{K}\Big(\prod_{h=1}^{m_{ij}}\Big(\rho_{kl}(t_{ijh})\cdot\frac{1}{G_l^{ijh}}\Big)\Big)^{z_{ik}z_{jl}}\Big|\mathbf{T};\Theta^*\Big]\\
&=\sum_{1\le i\neq j\le n}\mathbf{E}\Big[\sum_{k=1}^{K}\sum_{l=1}^{K}\Big(z_{ik}z_{jl}\cdot\sum_{h=1}^{m_{ij}}\big(\log\rho_{kl}(t_{ijh})-\log G_l^{ijh}\big)\Big)\Big|\mathbf{T};\Theta^*\Big]\\
&=\sum_{1\le i\neq j\le n}\sum_{k=1}^{K}\sum_{l=1}^{K}\Big(\mathbf{E}[z_{ik}z_{jl}|\mathbf{T};\Theta^*]\cdot\sum_{h=1}^{m_{ij}}\log\rho_{kl}(t_{ijh})\Big)\\
&\qquad\qquad-\sum_{1\le i\neq j\le n}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbf{E}\Big[z_{ik}z_{jl}\cdot\sum_{h=1}^{m_{ij}}\log G_l^{ijh}\Big|\mathbf{T};\Theta^*\Big].
\end{aligned}$$

The $\mathcal{L}^{P_2}$ part is equal to

$$
\begin{aligned}
&\sum_{i=1}^{n} \mathbf{E}\Big[\log \mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{Z})\Big|\mathbf{T};\Theta^*\Big] \qquad (A.4)\\
&= \sum_{i=1}^{n} \mathbf{E}\Big[\log \prod_{k=1}^{K}\Big(\prod_{h=1}^{M_i}\exp\Big(-\sum_{l=1}^{K}\int_{t_{ih}^-}^{t_{ih}} \rho_{kl}(t)\cdot I(G_l^{ih}>0)\mathrm{d}t\Big)\Big)^{z_{ik}}\Big|\mathbf{T};\Theta^*\Big]\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbf{E}\Big[z_{ik}\sum_{h=1}^{M_i}\Big(-\sum_{l=1}^{K}\int_{t_{ih}^-}^{t_{ih}} \rho_{kl}(t)\cdot I(G_l^{ih}>0)\mathrm{d}t\Big)\Big|\mathbf{T};\Theta^*\Big]\\
&= -\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{h=1}^{M_i}\sum_{l=1}^{K} \mathbf{E}\Big[z_{ik}\int_{t_{ih}^-}^{t_{ih}} \rho_{kl}(t)\cdot I(G_l^{ih}>0)\mathrm{d}t\Big|\mathbf{T};\Theta^*\Big]\\
&= -\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{h=1}^{M_i} \Big(\mathbf{E}\Big[z_{ik}I(G_l^{ih}>0)\Big|\mathbf{T};\Theta^*\Big]\cdot\int_{t_{ih}^-}^{t_{ih}} \rho_{kl}(t)\mathrm{d}t\Big),
\end{aligned}
$$

where we have pulled the indicator term $I(G_l^{ih}>0)$ out of the integral in the last step of (A.4) because the quantity G_l^{ih} is a constant on any $(t_{ih}^-, t_{ih}]$, as no player substitution can happen during that time. Finally, the $\mathcal{L}^O$ part is equal to

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \mathbf{E}\Big[\log \mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z})\Big|\mathbf{T};\Theta^*\Big] \qquad (A.5)\\
&= \sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \mathbf{E}\Big[\log \prod_{k=1}^{K}\Big(\prod_{h=1}^{m_{ia}}\eta_{ka}(t_{iah})\cdot\prod_{h=1}^{M_i}\exp\Big(-\int_{t_{ih}^-}^{t_{ih}} \eta_{ka}(t)\mathrm{d}t\Big)\Big)^{z_{ik}}\Big|\mathbf{T};\Theta^*\Big]\\
&= \sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\sum_{k=1}^{K} \mathbf{E}\Big[z_{ik}\Big(\sum_{h=1}^{m_{ia}}\log\eta_{ka}(t_{iah})-\sum_{h=1}^{M_i}\int_{t_{ih}^-}^{t_{ih}} \eta_{ka}(t)\mathrm{d}t\Big)\Big|\mathbf{T};\Theta^*\Big]\\
&= \sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\sum_{k=1}^{K} \Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\Big(\sum_{h=1}^{m_{ia}}\log\eta_{ka}(t_{iah})-\sum_{h=1}^{M_i}\int_{t_{ih}^-}^{t_{ih}} \eta_{ka}(t)\mathrm{d}t\Big)\Big),
\end{aligned}
$$

and the $\mathcal{L}(Z)$ part is equal to

$$\begin{aligned}\mathbf{E}\Big[\log \mathcal{L}(\mathbf{Z})|\mathbf{T};\Theta^*\Big] &= \mathbf{E}\Big[\log \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_k^{z_{ik}}|\mathbf{T};\Theta^*\Big] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\log\pi_k\Big).\end{aligned} \tag{A.6}$$

A.2 Analytic Updates of Marginal Probabilities and Initial Probabilities

In the conditional expectation of the log-likelihood (A.1), the term that contains $\mathbf{P}$, the transition probabilities from initial actions to players in different groups, appears in (A.2). It is

$$\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{si}\log P_{sk}\Big) \tag{A.7}$$

but there is a constraint

$$\sum_{k=1}^{K} P_{sk} = 1 \text{ for any } s\in\mathcal{S}. \tag{A.8}$$

Introducing Lagrange multipliers ζ_s, for each $s\in\mathcal{S}$, we get

$$\sum_{s\in\mathcal{S}}\Bigg[\sum_{i=1}^{n}\sum_{k=1}^{K}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{si}\log P_{sk}\Big) - \zeta_s\Big(\sum_{k=1}^{K}P_{sk}-1\Big)\Bigg]. \tag{A.9}$$

Differentiating with respect to each P_{sk} and setting the the derivatives to zero, we get

$$\frac{\sum_{i=1}^{n}\big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{si}\big)}{P_{sk}} - \zeta_s = 0, \text{ for } s\in\mathcal{S} \text{ and } k=1,2,\ldots,K. \tag{A.10}$$

The constraint (A.8) implies

$$\zeta_s = \sum_{k=1}^{K} \sum_{i=1}^{n} \big(\mathbf{E}[z_{ik} | \mathbf{T}; \Theta^*] \cdot m_{si} \big). \tag{A.11}$$

Hence, we obtain the updating equation (3.26):

$$P_{sk} = \frac{\sum_{i=1}^{n} \big(\mathbf{E}[z_{ik} | \mathbf{T}; \Theta^*] \cdot m_{si} \big)}{\sum_{k=1}^{K} \sum_{i=1}^{n} \big(\mathbf{E}[z_{ik} | \mathbf{T}; \Theta^*] \cdot m_{si} \big)}. \tag{A.12}$$

The updating equation (3.20) for $(\pi_1, \pi_2, \ldots, \pi_K)$ can be derived in a similar manner; the actual derivation is omitted.

A.3 $\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z}) | \mathbf{T}; \Theta^*]$ under Model Simplifications (3.21)-(3.22)

In Section 5, we introduced further simplifications to our Continuous-time SBM, namely (3.21) and (3.22), before applying it to analyze basketball games. Here, we provide details about the changes to some of the components (A.2)-(A.6) for $\mathbf{E}[\log \mathcal{L}(\mathbf{T}, \mathbf{Z}) | \mathbf{T}; \Theta^*]$ as a result of these simplifications. The components (A.2) and (A.6) do not involve any rate functions, so they remain the same; whereas the components (A.3)-(A.5) now become

$$\begin{aligned} &\sum_{1 \le i \ne j \le n} \mathbf{E}\Big[\log \mathcal{L}^{P_1}(\mathbf{T}_{ij} | \mathbf{Z}) \Big| \mathbf{T}; \Theta^* \Big] \\ &= \sum_{1 \le i \ne j \le n} \sum_{k=1}^{K} \sum_{l=1}^{K} \Big(\mathbf{E}[z_{ik} z_{jl} | \mathbf{T}; \Theta^*] \cdot \big(\sum_{h=1}^{m_{ij}} \log \lambda_k(t_{ijh}) + m_{ij} \log P_{kl} \big) \Big) \end{aligned} \tag{A.13}$$

$$- \sum_{1\le i\neq j\le n} \sum_{k=1}^{K}\sum_{l=1}^{K} \mathbf{E}\Big[z_{ik}z_{jl} \cdot \sum_{h=1}^{m_{ij}} \log G_l^{ijh} \Big| \mathbf{T}; \Theta^* \Big],$$

$$\begin{aligned} &\sum_{i=1}^{n} \mathbf{E}\Big[\log \mathcal{L}^{P_2}(\mathbf{T}_i|\mathbf{Z}) \Big| \mathbf{T}; \Theta^* \Big] \qquad (A.14)\\ &= -\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{h=1}^{M_i} \Big(\mathbf{E}\Big[z_{ik} I(G_l^{ih} > 0) \Big| \mathbf{T}; \Theta^* \Big] \cdot P_{kl} \cdot \int_{t_{ih}^-}^{t_{ih}} \lambda_k(t) \mathrm{d}t \Big), \end{aligned}$$

and

$$\begin{aligned} \sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \mathbf{E}\Big[\log \mathcal{L}^{O}(\mathbf{T}_{ia}|\mathbf{Z}) \Big| \mathbf{T}; \Theta^* \Big] = \sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\sum_{k=1}^{K} \Big(&\mathbf{E}[z_{ik}|\mathbf{T}; \Theta^*] \cdot \\ &\Big(\sum_{h=1}^{m_{ia}} \log \lambda_k(t_{iah}) + m_{ia} \log P_{ka} - P_{ka} \cdot \sum_{h=1}^{M_i} \int_{t_{ih}^-}^{t_{ih}} \lambda_k(t) \mathrm{d}t \Big) \Big). \qquad (A.15) \end{aligned}$$

A.4 Analytic Updates of Transition Probabilities under Model Simplifications (3.21)-(3.22)

Recall that, under model simplifications (3.21)-(3.22), the constraint on these transition probabilities is given by (3.23):

$$\sum_{l=1}^{K} P_{kl} + \sum_{a\in\mathcal{A}} P_{ka} = 1, \text{for any } k = 1, 2, \ldots, K. \qquad (A.16)$$

Again, we introduce Lagrange multiplier ζ_k for $k = 1, 2, \ldots, K$. Combining the terms from (A.13)-(A.15) that involve these transition probabilities with the constraint above, we obtain the Lagrangian function,

$$\begin{aligned}
&\sum_{1\le i\neq j\le n}\sum_{k=1}^{K}\sum_{l=1}^{K}\Big(\mathbf{E}[z_{ik}z_{jl}|\mathbf{T};\Theta^*]\cdot m_{ij}\cdot\log P_{kl}\Big) \\
&-\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{h=1}^{M_i}\Big(\mathbf{E}\Big[z_{ik}I(G_l^{ih}>0)\Big|\mathbf{T};\Theta^*\Big]\cdot P_{kl}\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\Big) \\
&+\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\sum_{k=1}^{K}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\big(m_{ia}\cdot\log P_{ka}-P_{ka}\cdot\sum_{h=1}^{M_i}\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\big)\Big) \\
&\qquad\qquad -\sum_{k=1}^{K}\zeta_k\cdot\Big(\sum_{l=1}^{K}P_{kl}+\sum_{a\in\mathcal{A}}P_{ka}-1\Big).
\end{aligned} \tag{A.17}$$

Differentiating with respect to each P_{kl}, P_{ka} and setting the the derivatives to zero, we get

$$\begin{aligned}
&\frac{\sum_{1\le i\neq j\le n}\Big(\mathbf{E}[z_{ik}z_{jl}|\mathbf{T};\Theta^*]\cdot m_{ij}\Big)}{P_{kl}} \\
&-\sum_{i=1}^{n}\sum_{h=1}^{M_i}\Big(\mathbf{E}\Big[z_{ik}I(G_l^{ih}>0)\Big|\mathbf{T};\Theta^*\Big]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\Big)-\zeta_k=0,
\end{aligned} \tag{A.18}$$

and

$$\begin{aligned}
&\frac{\sum_{i=1}^{n}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{ia}\Big)}{P_{ka}} \\
&\qquad -\sum_{i=1}^{n}\sum_{h=1}^{M_i}\Big(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\Big)-\zeta_k=0,
\end{aligned} \tag{A.19}$$

from which we can solve for the transition probabilities:

$$P_{kl} = \frac{\sum_{1\le i\ne j\le n}\left(\mathbf{E}[z_{ik}z_{jl}|\mathbf{T};\Theta^*]\cdot m_{ij}\right)}{\sum_{i=1}^{n}\sum_{h=1}^{M_i}\left(\mathbf{E}\left[z_{ik}I(G_l^{ih}>0)\middle|\mathbf{T};\Theta^*\right]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\right)+\zeta_k} \tag{A.20}$$

$$P_{ka} = \frac{\sum_{i=1}^{n}\left(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot m_{ia}\right)}{\sum_{i=1}^{n}\sum_{h=1}^{M_i}\left(\mathbf{E}[z_{ik}|\mathbf{T};\Theta^*]\cdot\int_{t_{ih}^-}^{t_{ih}}\lambda_k(t)\mathrm{d}t\right)+\zeta_k} \tag{A.21}$$

for $k,l=1,2,\ldots,K$ and $a\in\mathcal{A}$. Each Lagrange multiplier ζ_k can be solved numerically as the (univariate) root to the equation $\sum_{l=1}^{K}P_{kl}+\sum_{a\in\mathcal{A}}P_{ka}=1$ for each k. We do this with the R function `uniroot`.

A.5 Confidence Bands for Estimated Rate Functions

It is possible to obtain confidence bands for the estimated rate functions conditional on the cluster labels by calculating the pointwise standard errors using the observed Fisher information matrix and the standard Delta method. As an example, rate functions displayed on top of each other in Figure 3.4 (to facilitate side-by-side comparison in Section 3.3.2) are now displayed individually in Figure A.1 with their respective 95% confidence bands. In all panels of Figure A.1, we can see that, as $t\to 24$, the confidence intervals invariably widen. This is because there are fewer transactions as the time approaches the end limit for each play, since many plays end before reaching the full 24-second limit. Elsewhere, these confidence intervals are narrow enough to suggest that features identified in Figure 3.4 and discussed in Section 3.3.2 are unlikely to be merely artifacts due to noise in the data.

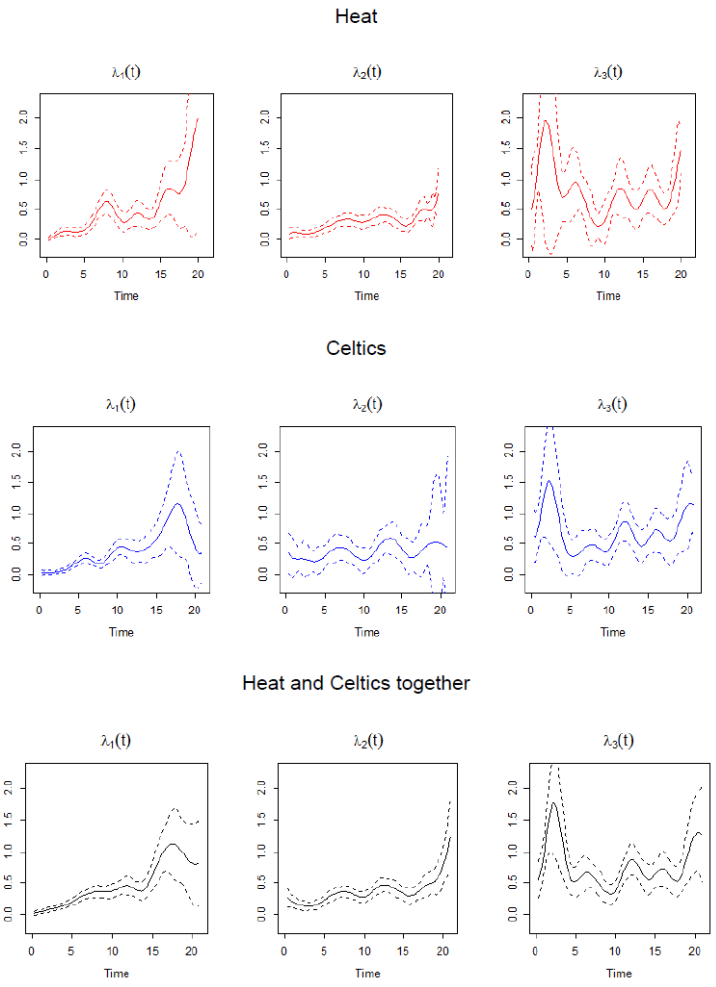


Figure A.1: Rate functions displayed on top of each other in Figure 3.4 are displayed here individually with 95% pointwise confidence bands.

Appendix B

Proofs for Section 4.2

B.1 Proof of Lemma 1

First, it is easy to show

$$\mathcal{S}(X_{-i}) = \mathcal{S}(X_{-ij}) + \mathcal{S}(M_{ij}X_j), \tag{B.1}$$

where $M_{ij}X_j$ is the projection of X_j into the space $\mathcal{S}^{\perp}(X_{-ij})$. In fact, the column vector X_j can be decomposed as

$$\begin{aligned} X_j &= P(X_{-ij})X_j + (I_n - P(X_{-ij}))X_j \\ &= P(X_{-ij})X_j + M_{ij}X_j. \end{aligned} \tag{B.2}$$

On the one hand, $P(X_{-ij})X_j \in \mathcal{S}(X_{-ij})$, so $\mathcal{S}(X_j) \subseteq \mathcal{S}(X_{-ij}) + \mathcal{S}(M_{ij}X_j)$, and thus,

$$\begin{aligned} \mathcal{S}(X_{-i}) &\subseteq \mathcal{S}(X_{-ij}) + \mathcal{S}(X_j) \\ &\subseteq \mathcal{S}(X_{-ij}) + \mathcal{S}(M_{ij}X_j). \end{aligned} \tag{B.3}$$

On the other hand, $M_{ij}X_j = X_j - P(X_{-ij})X_j \in \mathcal{S}(X_j) + \mathcal{S}(X_{-ij})$, then,

$$\begin{aligned} \mathcal{S}(X_{-ij}) + \mathcal{S}(M_{ij}X_j) &\subseteq \mathcal{S}(X_{-ij}) + \mathcal{S}(X_j) \\ &\subseteq \mathcal{S}(X_{-i}). \end{aligned} \tag{B.4}$$

Hence, combining (B.3) and (B.4) yields (B.1).

Second, clearly $\mathcal{S}(X_{-ij})$ and $\mathcal{S}(M_{ij}X_j)$ are orthogonal subspaces of $\mathcal{S}(X_{-i})$, so

$$P(X_{-i}) = P(X_{-ij}) + P(M_{ij}X_j) \text{ and } P(X_{-ij})P(M_{ij}X_j) = 0. \tag{B.5}$$

Then,

$$I_n - P(X_{-i}) = I_n - P(X_{-ij}) - P(M_{ij}X_j), \tag{B.6}$$

which is

$$M_i = M_{ij} - P(M_{ij}X_j), \tag{B.7}$$

and

$$M_i P(M_{ij}X_j) = 0. \tag{B.8}$$

Due to (B.7),

$$M_{ij}X_{ij} = (M_i + P(M_{ij}X_j))X_{ij} \tag{B.9}$$

$$= (M_i + P(M_{ij}X_j))(X_i \quad X_j) \tag{B.10}$$

$$= (M_iX_i + P(M_{ij}X_j)X_i \quad M_iX_j + P(M_{ij}X_j)X_j) \tag{B.11}$$

$$= (M_iX_i + P(M_{ij}X_j)X_i \quad M_{ij}X_j), \tag{B.12}$$

where the third equation is because $M_iX_j = 0$ and $P(M_{ij}X_j)X_j = M_{ij}X_j$, which can be easily proven. Finally, the noncentrality parameter

$$\Lambda_{ij} = \frac{1}{\sigma^2}(\beta_i, \beta_j)X_{ij}^T M_{ij} X_{ij}(\beta_i, \beta_j)^T$$

(M_{ij} is an idempotent matrix and is symmetric)

$$= \frac{1}{\sigma^2}(\beta_i, \beta_j)X_{ij}^T M_{ij}^T M_{ij} X_{ij}(\beta_i, \beta_j)^T$$

$$= \frac{1}{\sigma^2}||M_{ij}X_{ij}\beta||^2 \tag{B.13}$$

$$= \frac{1}{\sigma^2}||M_iX_i\beta_i + P(M_{ij}X_j)X_i\beta_i + M_{ij}X_j\beta_j||^2 \tag{B.14}$$

$$= \frac{1}{\sigma^2}\left(||M_iX_i\beta_i||^2 + ||P(M_{ij}X_j)X_i\beta_i + M_{ij}X_j\beta_j||^2\right) \tag{B.15}$$

$$\geq \frac{1}{\sigma^2}||M_iX_i\beta_i||^2, \tag{B.16}$$

where $||\cdot||$ denotes the L2-norm of a vector. The last equation holds because $X_i^T M_i P(M_{ij}X_j)$=0 and $X_i^T M_i M_{ij} X_j = 0$, which is due to (B.8).

Finally, utilize assumptions (A1) and (A2), for any i that $\beta_i \neq 0$, as $n \to \infty$,

$$\frac{\Lambda_{ij}}{\log n} \geq \frac{||M_i X_i \beta_i||^2}{\sigma^2 \log n} = \frac{\beta_i^2 X_i^T M_i X_i}{\sigma^2 \log n} \geq \frac{\psi^2 X_i^T M_i X_i}{\sigma^2 \log n} \to \infty. \tag{B.17}$$

We now show a side result that is used in Section 4.2.2. From (B.12), we have

$$\mathcal{S}(M_{ij} X_{ij}) = \mathcal{S}(M_i X_i) + \mathcal{S}(M_{ij} X_j). \tag{B.18}$$

In addition, the subspaces $\mathcal{S}(M_i X_i)$ and $\mathcal{S}(M_{ij} X_j)$ are orthogonal because of (B.8). Therefore, the projection matrix $P(M_{ij} X_{ij})$ can be written as

$$P(M_{ij} X_{ij}) = P(M_i X_i) + P(M_{ij} X_j), \tag{B.19}$$

and

$$P(M_i X_i) P(M_{ij} X_j) = 0. \tag{B.20}$$

B.2 Proof of Lemma 2

Recall that,

$$\alpha_{ij} = \mathbf{P}\big(F(2, n-p, \Lambda_{ij}) > C_{\alpha_n}\big), \tag{B.21}$$

where C_{α_n} is the $1 - \alpha_n$ quantile of $F(2, n-p)$.

Suppose $q(\alpha, k)$ is the $1-\alpha$ quantile of $\chi^2(k)$, i.e.,

$$\mathbf{P}(\chi^2(k) > q(\alpha, k)) = \alpha.$$

We will prove the Lemma in two steps.

Step 1: we show that if $\lim_{n\to\infty} \alpha_n n^2 = \infty$, when $n-p$ is big enough, we have $C_{\alpha_n} < q(\alpha_n/2, 2)$.

By the definition of C_{α_n},

$$\mathbf{P}\big(F(2, n-p) > C_{\alpha_n}\big) = \alpha_n. \tag{B.22}$$

We will show, for big enough $n-p$,

$$\mathbf{P}\Big(F(2, n-p) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) < \alpha_n, \tag{B.23}$$

which implies that $C_{\alpha_n} < q(\alpha_n/2, 2)$.

In fact,

$$\mathbf{P}\Big(F(2, n-p) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) \tag{B.24}$$

$$= \mathbf{P}\Big(\frac{\chi^2(2)/2}{\chi^2(n-p)/(n-p)} > q\big(\frac{\alpha_n}{2}, 2\big)\Big) \tag{B.25}$$

$$= \mathbf{P}\Big(\frac{\chi^2(2)/2}{\chi^2(n-p)/(n-p)} > q\big(\frac{\alpha_n}{2}, 2\big)\Big|\frac{\chi^2(n-p)}{n-p} > \frac{1}{2}\Big) \cdot \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} > \frac{1}{2}\Big) \tag{B.26}$$

$$+ \mathbf{P}\Big(\frac{\chi^2(2)/2}{\chi^2(n-p)/(n-p)} > q\big(\frac{\alpha_n}{2}, 2\big)\Big|\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) \cdot \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) \tag{B.27}$$

$$< \mathbf{P}\Big(\chi^2(2) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) + \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big). \tag{B.28}$$

By definition, the first term $\mathbf{P}\Big(\chi^2(2) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) = \frac{\alpha_n}{2}$.

Now we look at the second term

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \leq \frac{1}{2}\Big) = \mathbf{P}\Big(\sqrt{\frac{n-p}{2}}\big(\frac{\chi^2(n-p)}{n-p} - 1\big) \leq -\sqrt{\frac{n-p}{8}}\Big). \tag{B.29}$$

The nonuniform Berry-Esséen bound (Michel, 1981; Chen and Shao, 2001) is a standard bound for the convergence rate of the Central Limit Theorem. It states that for all $n \in \mathbb{N}$, i.i.d. random variables $X_1, X_2, \cdots, X_n$ with mean μ and variance σ^2, and $\mathbf{E}|X_i|^3 < \infty$,

$$\Big|\mathbf{P}\Big(\frac{\sqrt{n}}{\sigma}\big(\frac{\sum_{i=1}^n X_i}{n} - \mu\big) < x\Big) - \Phi(x)\Big| \leq \frac{C}{\sqrt{n}(1+|x|^3)}, \tag{B.30}$$

where $\Phi(x)$ is the CDF of the standard normal distribution, and C is a constant. Applying the bound to (B.29), we have

$$\Big|\mathbf{P}\Big(\sqrt{\frac{n-p}{2}}\big(\frac{\chi^2(n-p)}{n-p} - 1\big) \leq -\sqrt{\frac{n-p}{8}}\Big) - \Phi(-\sqrt{\frac{n-p}{8}})\Big| \tag{B.31}$$

$$\leq \frac{16\sqrt{2}C}{\sqrt{n-p}(16\sqrt{2} + (n-p)^{\frac{3}{2}})} \tag{B.32}$$

$$< \frac{16\sqrt{2}C}{(n-p)^2} \tag{B.33}$$

$$< \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2}, \tag{B.34}$$

where the last inequality is due to the settings $p < \gamma n$.

Therefore,

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) = \mathbf{P}\Big(\sqrt{\frac{n-p}{2}}\big(\frac{\chi^2(n-p)}{n-p} - 1\big) \le -\sqrt{\frac{n-p}{8}}\Big) \tag{B.35}$$

$$< \Phi\Big(-\sqrt{\frac{n-p}{8}}\Big) + \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2} \tag{B.36}$$

$$= 1 - \Phi\Big(\sqrt{\frac{n-p}{8}}\Big) + \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2}. \tag{B.37}$$

Apply the standard tail bound for $N(0,1)$: $1 - \Phi(x) \le x^{-1}e^{-x^2/2}$

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) < \sqrt{\frac{8}{n-p}}\exp\big(-\frac{n-p}{16}\big) + \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2}. \tag{B.38}$$

Again, due to the settings $p < \gamma n$ and $\alpha_n n^2 \to \infty$, for big enough n,

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) < \exp\big(-\frac{(1-\gamma)n}{16}\big) + \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2} \tag{B.39}$$

$$< \frac{\alpha_n}{2}. \tag{B.40}$$

Finally, according to (B.28),

$$\mathbf{P}\Big(F(2, n-p) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) \tag{B.41}$$

$$< \mathbf{P}\Big(\chi^2(2) > q\big(\frac{\alpha_n}{2}, 2\big)\Big) + \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \le \frac{1}{2}\Big) \tag{B.42}$$

$$< \frac{\alpha_n}{2} + \frac{\alpha_n}{2} \tag{B.43}$$

$$= \alpha_n, \tag{B.44}$$

which completes the proof of **Step 1**.

Step 2: we prove the conclusion of the Lemma. In step 1, we have shown that, when $n-p$ is big enough, $C_{\alpha_n} < q(\alpha_n/2, 2)$, so

$$1 - \alpha_{ij} = \mathbf{P}\big(F(2, n-p, \Lambda_{ij}) \le C_{\alpha_n}\big) \tag{B.45}$$

$$< \mathbf{P}\big(F(2, n-p, \Lambda_{ij}) \le q(\frac{\alpha_n}{2}, 2)\big). \tag{B.46}$$

Meanwhile,

$$\mathbf{P}\Big(F(2, n-p, \Lambda_{ij}) \le q\big(\frac{\alpha_n}{2}, 2\big)\Big) \tag{B.47}$$

$$= \mathbf{P}\Big(\frac{\chi^2(2, \Lambda_{ij})/2}{\chi^2(n-p)/(n-p)} \le q\big(\frac{\alpha_n}{2}, 2\big)\Big) \tag{B.48}$$

$$= \mathbf{P}\Big(\frac{\chi^2(2, \Lambda_{ij})/2}{\chi^2(n-p)/(n-p)} \le q\big(\frac{\alpha_n}{2}, 2\big)\Big|\frac{\chi^2(n-p)}{n-p} < \frac{3}{2}\Big) \cdot \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} < \frac{3}{2}\Big) \tag{B.49}$$

$$+ \mathbf{P}\Big(\frac{\chi^2(2, \Lambda_{ij})/2}{\chi^2(n-p)/(n-p)} \le q\big(\frac{\alpha_n}{2}, 2\big)\Big|\frac{\chi^2(n-p)}{n-p} \ge \frac{3}{2}\Big) \cdot \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \ge \frac{3}{2}\Big) \tag{B.50}$$

$$< \mathbf{P}\Big(\chi^2(2, \Lambda_{ij}) \le 3q\big(\frac{\alpha_n}{2}, 2\big)\Big) + \mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \ge \frac{3}{2}\Big). \tag{B.51}$$

We will show that both terms in the last line above are $o(\frac{1}{n})$, so $1 - \alpha_{ij} = o(\frac{1}{n})$, which completes the proof of the lemma.

We first look at the second term,

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \ge \frac{3}{2}\Big) = \mathbf{P}\Big(\sqrt{\frac{n-p}{2}}\big(\frac{\chi^2(n-p)}{n-p} - 1\big) \ge \sqrt{\frac{n-p}{8}}\Big). \tag{B.52}$$

Following very similar arguments from (B.29) to (B.39), we obtain

$$\mathbf{P}\Big(\frac{\chi^2(n-p)}{n-p} \geq \frac{3}{2}\Big) < \exp\big(-\frac{(1-\gamma)n}{16}\big) + \frac{16\sqrt{2}C}{(1-\gamma)^2 n^2} = o(\frac{1}{n}). \tag{B.53}$$

To investigate the first term $\mathbf{P}\Big(\chi^2(2,\Lambda_{ij}) \leq 3q\big(\frac{\alpha_n}{2},2\big)\Big)$, we utilize an upper bound of $q(\alpha,k)$, i.e., the $1-\alpha$ quantile of $\chi^2(k)$, given by Laurent and Massart (2000): for any $\alpha \in (0,1)$ and integer $k>0$,

$$q(\alpha,k) \leq k + 2\log\big(\frac{1}{\alpha}\big) + 2\sqrt{k\log\big(\frac{1}{\alpha}\big)}. \tag{B.54}$$

Also note that for any $x>0$, $\Lambda>0$ and two integers $k_2>k_1>0$,

$$\mathbf{P}(\chi^2(k_2,\Lambda) \leq x) < \mathbf{P}(\chi^2(k_1,\Lambda) \leq x). \tag{B.55}$$

Hence,

$$\mathbf{P}\Big(\chi^2(2,\Lambda_{ij}) \leq 3q\big(\frac{\alpha_n}{2},2\big)\Big) \tag{B.56}$$

$$\leq \mathbf{P}\Big(\chi^2(2,\Lambda_{ij}) \leq 6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}\Big) \tag{B.57}$$

$$< \mathbf{P}\Big(\chi^2(1,\Lambda_{ij}) \leq 6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}\Big) \tag{B.58}$$

$$= \mathbf{P}\Big(|N(\sqrt{\Lambda_{ij}},1)| \leq \sqrt{6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}}\Big) \tag{B.59}$$

$$< \mathbf{P}\Big(N(\sqrt{\Lambda_{ij}},1) \leq \sqrt{6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}}\Big) \tag{B.60}$$

$$= \mathbf{P}\Big(N(0,1) \le \sqrt{6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}} - \sqrt{\Lambda_{ij}}\Big). \tag{B.61}$$

Since $\alpha_n n^2 \to \infty$, when n is big enough, we have $\alpha_n n^2 \ge 2$, i.e. $\frac{2}{\alpha_n} \le n^2$. Therefore,

$$\sqrt{6 + 6\log\big(\frac{2}{\alpha_n}\big) + 6\sqrt{2\log\big(\frac{2}{\alpha_n}\big)}} \le \sqrt{6 + 12\log n + 12\sqrt{\log n}}. \tag{B.62}$$

Due to the assumption $\Lambda_{ij}/\log n \to \infty$, it is easy to see that, for large enough n,

$$\sqrt{6 + 12\log n + 12\sqrt{\log n}} - \sqrt{\Lambda_{ij}} < -\frac{\sqrt{\Lambda_{ij}}}{2}. \tag{B.63}$$

Consequently, as $n \to \infty$

$$\mathbf{P}\Big(\chi^2(2,\Lambda_{ij}) \le 3q\big(\frac{\alpha_n}{2}, 2\big)\Big) < \mathbf{P}\Big(N(0,1) \le -\frac{\sqrt{\Lambda_{ij}}}{2}\Big) \tag{B.64}$$

$$= 1 - \Phi\big(\frac{\sqrt{\Lambda_{ij}}}{2}\big) \tag{B.65}$$

$$\le \frac{2}{\sqrt{\Lambda_{ij}}} \exp\big(-\frac{\Lambda_{ij}}{8}\big) \tag{B.66}$$

$$\le \frac{2}{\sqrt{\Lambda_{ij}}} \cdot \frac{1}{n} \tag{B.67}$$

$$= o(\frac{1}{n}). \tag{B.68}$$

Therefore, both terms in (B.51) are $o(\frac{1}{n})$, which implies $1 - \alpha_{ij} = o(\frac{1}{n})$.

B.3 Proof of Theorem 1

For a relevant variable i, i.e., $i \in D$, due to Lemma 1 and 2, we have $1 - \alpha_{ij} = o(\frac{1}{n})$, for all $j \neq i$, so

$$\begin{aligned}
\mathbf{E}\big(|p - 1 - d(i)|\big) &= p - 1 - \sum_{k \neq i} \mathbf{E}(A_{ik}) \\
&= \sum_{k \neq i} (1 - \alpha_{ik}) \\
&= o\Big(\frac{p-1}{n}\Big) \qquad \text{(B.69)} \\
&= o(1). \qquad \text{(B.70)}
\end{aligned}$$

According to Markov inequality,

$$\mathbf{P}\Big(|d(i) - (p-1)| > \frac{(1-\delta)p}{4}\Big) \leq \frac{4 \cdot \mathbf{E}(|d(i) - (p-1)|)}{(1-\delta)p} = o\Big(\frac{1}{p}\Big). \qquad \text{(B.71)}$$

Therefore,

$$\mathbf{P}\Big(\max_{i \in D} |d(i) - (p-1)| > \frac{(1-\delta)p}{4}\Big) \leq o\Big(\frac{s}{p}\Big) = o(1). \qquad \text{(B.72)}$$

For an irrelevant variable j, i.e., $j \notin D$, also due to Lemma 1 and 2, we have $1 - \alpha_{jk} = o(\frac{1}{n})$, for all $k \in D$; and we let $\alpha_n = o(\frac{1}{p})$, so

$$\begin{aligned}
\mathbf{E}\big(|d(j) - s|\big) &= s - \sum_{k \neq j} \mathbf{E}(A_{jk}) + \sum_{k \neq j, k \notin D} \mathbf{E}(A_{jk}) \\
&= \sum_{k \in D} (1 - \alpha_{jk}) + \sum_{k \neq j, k \notin D} \alpha_n
\end{aligned}$$

$$= o\Big(\frac{s}{n}\Big) + o\Big(\frac{p-s-1}{p}\Big) \tag{B.73}$$

$$= o(1). \tag{B.74}$$

Again, according to Markov inequality,

$$\mathbf{P}(|d(j)-s| > \frac{(1-\delta)p}{4}) \leq \frac{4 \cdot \mathbf{E}(|d(j)-s|)}{(1-\delta)p} = o\Big(\frac{1}{p}\Big). \tag{B.75}$$

Therefore,

$$\mathbf{P}(\max_{j \notin D} |d(j)-s| > \frac{(1-\delta)p}{4}) \leq o\Big(\frac{p-s}{p}\Big)$$

$$= o(1). \tag{B.76}$$

B.4 Proof of Lemma 3

$$\mathbf{P}(A_{ij}=1) = \mathbf{P}(TS_{ij} > C_{\alpha_n}) \tag{B.77}$$

$$= \mathbf{P}\Big(\frac{(B_i + C_{ij})/2}{Y^T(I_n - P)Y/\sigma^2/(n-p)} > C_{\alpha_n}\Big) \tag{B.78}$$

$$= \mathbf{P}\Big(\frac{(B_i + C_{ij})}{Y^T(I_n - P)Y/\sigma^2/(n-p)} > 2C_{\alpha_n}\Big). \tag{B.79}$$

We will show that, if $n-p \to \infty$ and $\alpha_n^c p \to \infty$, for a constant $c > 2$, we have

$$\mathbf{P}\Big(\max_{i \notin D} \frac{B_i}{Y^T(I_n - P)Y/\sigma^2/(n-p)} > 2C_{\alpha_n}\Big) \to 1. \tag{B.80}$$

Therefore, there exists at least one irrelevant variable, say k, that

$$\mathbf{P}\Big(\frac{B_k}{Y^T(I_n-P)Y/\sigma^2/(n-p)} > 2C_{\alpha_n}\Big) \to 1. \tag{B.81}$$

Since $C_{kj} \geq 0$, for all $j \neq k$, we obtain

$$\mathbf{P}\Big(\min_{j\neq k} TS_{kj} > C_{\alpha_n}\Big) = \mathbf{P}\Big(\min_{j\neq k} \frac{B_k + C_{kj}}{Y^T(I_n-P)Y/\sigma^2/(n-p)} > 2C_{\alpha_n}\Big) \to 1, \tag{B.82}$$

which implies that the irrelevant variable k has a probability tending to 1 to connect to all the other variables.

Now we prove (B.80).

Let $G = Y^T(I_n - P)Y/\sigma^2$, so $G \sim \chi^2(n-p)$, and G and B_i are independent for all $i \notin D$. Then,

$$\mathbf{P}\Big(\max_{i\notin D} \frac{B_i}{Y^T(I_n-P)Y/\sigma^2/(n-p)} \leq 2C_{\alpha_n}\Big) \tag{B.83}$$

$$= \mathbf{P}\Big(\max_{i\notin D} \frac{B_i}{G/(n-p)} \leq 2C_{\alpha_n}\Big) \tag{B.84}$$

$$= \mathbf{P}\Big(\max_{i\notin D} \frac{B_i}{G/(n-p)} \leq 2C_{\alpha_n} \Big| \frac{G}{n-p} < \sqrt{\frac{c_1}{2}}\Big) \cdot \mathbf{P}\Big(\frac{G}{n-p} < \sqrt{\frac{c_1}{2}}\Big) \tag{B.85}$$

$$+ \mathbf{P}\Big(\max_{i\notin D} \frac{B_i}{G/(n-p)} \geq 2C_{\alpha_n} \Big| \frac{G}{n-p} \geq \sqrt{\frac{c_1}{2}}\Big) \cdot \mathbf{P}\Big(\frac{G}{n-p} \geq \sqrt{\frac{c_1}{2}}\Big) \tag{B.86}$$

$$< \mathbf{P}(\max_{i\notin D} B_i \leq \sqrt{2c_1}C_{\alpha_n}) + \mathbf{P}\Big(\frac{G}{n-p} \geq \sqrt{\frac{c_1}{2}}\Big). \tag{B.87}$$

The constant $c_1 \in (2, c)$, where $c > 2$ is the constant in the condition that lets $\lim_{n\to\infty} \alpha_n^c p = \infty$.

We first look at the second term. According to the law of large numbers, $\frac{G}{n-p} \xrightarrow{a.s.} 1$, since $\sqrt{\frac{c_1}{2}} > 1$, we have, as $n - p \to \infty$,

$$\mathbf{P}\Big(\frac{G}{n-p} \geq \sqrt{\frac{c_1}{2}}\Big) \to 0. \tag{B.88}$$

Now we look at the first term. When the design matrix is orthogonal, we have $X_i^T M_i M_k X_k = X_i^T X_k = 0$ for all i and k, which implies $P(M_i X_i)P(M_k X_k) = 0$. Recall that $B_i = ||P(M_i X_i)Y||^2/\sigma^2$, so $\{B_i : i \notin D\}$ are independent $\chi^2(1)$. Hence,

$$\mathbf{P}(\max_{i \notin D} B_i \leq \sqrt{2c_1}C_{\alpha_n}) = \mathbf{P}(B_i \leq \sqrt{2c_1}C_{\alpha_n})^{p-s} < \mathbf{P}(B_i \leq \sqrt{2c_1}C_{\alpha_n})^{(1-\delta)p}, \tag{B.89}$$

where the last inequality is due to the assumption that $s < \delta p$ for a constant $\delta \in (0, 1)$.

It is easy to see that if $\alpha_n \not\to 0$, which means α_n is bounded in between 0 and 1, so $p \to \infty$ and $\mathbf{P}(B_i \leq \sqrt{2c_1}C_{\alpha_n})^{(1-\delta)p} \to 0$.

From now on, we consider the case where $\alpha_n \to 0$.

We have shown in the proof of Lemma 2 that if $\alpha_n n^2 \to \infty$ and $n - p \to \infty$, we have, for big enough n, $C_{\alpha_n} < q(\alpha_n/2, 2)$, where $q(\alpha_n/2, 2)$ is the $1 - \alpha_n/2$ quantile of $\chi^2(2)$. Applying the upper bound of chi-squared quantile (B.54), we obtain that

$$\mathbf{P}(B_i \leq \sqrt{2c_1}C_{\alpha_n})^{(1-\delta)p} \tag{B.90}$$

$$< \mathbf{P}\Big(B_i \leq \sqrt{2c_1}q(\frac{\alpha_n}{2}, 2)\Big)^{(1-\delta)p} \tag{B.91}$$

$$\leq \mathbf{P}\Big(B_i \leq \sqrt{2c_1}\big(2 + 2\log\frac{2}{\alpha_n} + 2\sqrt{2\log\frac{2}{\alpha_n}}\big)\Big)^{(1-\delta)p} \tag{B.92}$$

$$= \mathbf{P}\Big(B_i \le 2\sqrt{2c_1}\big(1 + \log\frac{2}{\alpha_n} + \sqrt{2\log\frac{2}{\alpha_n}}\big)\Big)^{(1-\delta)p} \tag{B.93}$$

$$< \mathbf{P}\Big(B_i \le 2\sqrt{2c_1}\cdot\sqrt{\frac{c_1}{2}}\log\frac{2}{\alpha_n}\Big)^{(1-\delta)p} \tag{B.94}$$

$$= \mathbf{P}\Big(B_i \le 2c_1\log\frac{2}{\alpha_n}\Big)^{(1-\delta)p}, \tag{B.95}$$

where the last inequality is true because $\sqrt{\frac{c_1}{2}} > 1$, and thus for small enough α_n,

$$1 + \log\frac{2}{\alpha_n} + \sqrt{2\log\frac{2}{\alpha_n}} < \sqrt{\frac{c_1}{2}}\log\frac{2}{\alpha_n}. \tag{B.96}$$

Applying $B_i \sim \chi^2(1)$, and the standard lower bound for the tail of $N(0,1)$: $1 - \Phi(x) \ge \frac{1}{\sqrt{2\pi}}\frac{x}{x^2+1}\exp(-\frac{x^2}{2})$, we get

$$\mathbf{P}\Big(B_i \le 2c_1\log\frac{2}{\alpha_n}\Big)^{(1-\delta)p} \tag{B.97}$$

$$= \mathbf{P}\Big(\chi^2(1) \le 2c_1\log\frac{2}{\alpha_n}\Big)^{(1-\delta)p} \tag{B.98}$$

$$= \mathbf{P}\Big(|N(0,1)| \le \sqrt{2c_1\log\frac{2}{\alpha_n}}\Big)^{(1-\delta)p} \tag{B.99}$$

$$= \mathbf{P}\Big(2\Phi\big(\sqrt{2c_1\log\frac{2}{\alpha_n}}\big) - 1\Big)^{(1-\delta)p} \tag{B.100}$$

$$\le \Big(1 - \frac{2}{\sqrt{2\pi}}\cdot\frac{\sqrt{2c_1\log\frac{2}{\alpha_n}}}{2c_1\log\frac{2}{\alpha_n}+1}\exp\big(-c_1\log\frac{2}{\alpha_n}\big)\Big)^{(1-\delta)p}. \tag{B.101}$$

For simplicity, let $L_n = \sqrt{2c_1\log\frac{2}{\alpha_n}}/(2c_1\log\frac{2}{\alpha_n}+1)$, then

$$\Big(1 - \frac{2}{\sqrt{2\pi}}\cdot\frac{\sqrt{2c_1\log\frac{2}{\alpha_n}}}{2c_1\log\frac{2}{\alpha_n}+1}\exp\big(-c_1\log\frac{2}{\alpha_n}\big)\Big)^{(1-\delta)p} \tag{B.102}$$

$$= \Big(1 - \frac{2}{\sqrt{2\pi}} \cdot L_n \exp\big(-c_1 \log \frac{2}{\alpha_n}\big)\Big)^{(1-\delta)p} \tag{B.103}$$

$$= \Big(1 - \frac{2}{\sqrt{2\pi}} \cdot L_n \big(\frac{\alpha_n}{2}\big)^{c_1}\Big)^{(1-\delta)p}. \tag{B.104}$$

Let $C = \frac{2^{1-c_1}}{\sqrt{2\pi}}$,

$$\Big(1 - \frac{2}{\sqrt{2\pi}} \cdot L_n \big(\frac{\alpha_n}{2}\big)^{c_1}\Big)^{(1-\delta)p} \tag{B.105}$$

$$= \Big(1 - CL_n\alpha_n^{c_1}\Big)^{(1-\delta)p} \tag{B.106}$$

$$= \Big(1 - CL_n\alpha_n^{c_1}\Big)^{\frac{1}{L_n\alpha_n^{c_1}} \cdot L_n\alpha_n^{c_1}(1-\delta)p}. \tag{B.107}$$

Since as $\alpha_n \to 0$, we have $L_n \to 0$ and $\alpha_n^{c_1} \to 0$, so

$$\Big(1 - CL_n\alpha_n^{c_1}\Big)^{\frac{1}{L_n\alpha_n^{c_1}}} \to e^{-C}. \tag{B.108}$$

Now we show that when $\lim_{n\to\infty} \alpha_n^c p = \infty$,

$$L_n\alpha_n^{c_1}(1-\delta)p = \frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1}\alpha_n^{c_1}(1-\delta)p \to \infty. \tag{B.109}$$

In fact,

$$\frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1}\alpha_n^{c_1}(1-\delta)p = \frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1}\alpha_n^{c_1-c} \cdot \alpha_n^c(1-\delta)p. \tag{B.110}$$

Since $c_1 < c$, and $\frac{1}{\alpha_n} \to \infty$, clearly,

$$\frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1} \alpha_n^{c_1 - c} = \frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1} (\frac{1}{\alpha_n})^{c - c_1} \to \infty, \tag{B.111}$$

because $(\frac{1}{\alpha_n})^{c-c_1}$ goes to infinity in a polynomial rate of $\frac{1}{\alpha_n}$, while

$$\frac{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}{2c_1 \log \frac{2}{\alpha_n} + 1} \approx \frac{1}{\sqrt{2c_1 \log \frac{2}{\alpha_n}}}, \tag{B.112}$$

which goes to zero in a rate of $\sqrt{\log \frac{1}{\alpha_n}}$.

Together with the assumption that $\lim_{n \to \infty} \alpha_n^c p = \infty$, (B.109) is proven.

Hence, due to (B.108) and (B.109), we have shown that the first term of (B.87)

$$\mathbf{P}(\max_{i \notin D} B_i \leq \sqrt{2c_1} C_{\alpha_n}) = \mathbf{P}(B_i \leq \sqrt{2c_1} C_{\alpha_n})^{(1-\delta)p} \tag{B.113}$$

$$< \left(1 - C L_n \alpha_n^{c_1}\right)^{\frac{1}{L_n \alpha_n^{c_1}} \cdot L_n \alpha_n^{c_1} (1-\delta)p} \tag{B.114}$$

$$\to (e^{-C})^{\infty} = 0. \tag{B.115}$$

Finally, combining (B.87) and (B.88), we have proven (B.80)

$$\mathbf{P}\Big(\max_{i \notin D} \frac{B_i}{Y^T (I_n - P) Y / \sigma^2 / (n-p)} \leq 2 C_{\alpha_n} \Big) \to 0, \tag{B.116}$$

which completes the entire proof of Lemma 3.